AMSAA

TECHNICAL MEMORANDUM NO. 131

CONVERSION OF MICOM, TIME-PHASED, LIFE-CYCLE, COST-ESTIMATING MODEL FROM COBOL TO FORTRAN IV

Wayne S. Copes

March 1972

Approved for public release;
distribution unlimited.

U.S. ARMY MATERIEL SYSTEMS ANALYSIS AGENCY
Aberdeen Proving Ground, Maryland

DISPOSITION

Destroy this report when no longer needed. Do not return it to the originator.

DISCLAIMER

The findings in this report are not to be construed as an official Department of the Army position.

WARNING

Information and data contained in this document are based on the input available at the time of preparation. The results may be subject to change and should not be construed as representing the AMC position unless so specified.

TECHNICAL MEMORANDUM NO. 131

CONVERSION OF MICOM, TIME-PHASED, LIFE-CYCLE, COST-ESTIMATING MODEL FROM COBOL TO FORTRAN IV

Wayne S. Copes

March 1972

Approved for public release;
distribution unlimited.

RDT&E Project No. 1P765801MM11

U.S. ARMY MATERIEL SYSTEMS ANALYSIS AGENCY
ABERDEEN PROVING GROUND, MARYLAND

U.S. ARMY MATERIEL SYSTEMS ANALYSIS AGENCY

TECHNICAL MEMORANDUM NO. 131

WSCopes/sjj
Aberdeen Proving Ground, Md.
March 1972

CONVERSION OF MICOM, TIME-PHASED, LIFE-CYCLE, COST-ESTIMATING MODEL FROM COBOL TO FORTRAN IV

ABSTRACT

LICEM is a computer model which may be used to generate Time-Phased Life-Cycle Cost Estimates (LCCE) for personnel or materiel systems. The input to this model is in a form compatible with the Army Materiel Command's Improved Cost Estimating Project, Phase III (ICE-III).

The cost for a system can be computed for as many as thirty equal increments of time, and can be summarized in up to nine levels of complexity. The model estimates a system cost for each time increment as well as the total cost over the life of the system.

The model is written in FORTRAN IV specifically for the Ballistic Research Laboratories' Electronic Scientific Computer (BRLESC).

Next page is blank.

CONTENTS

Next page is blank.

CONVERSION OF MICOM, TIME-PHASED, LIFE-CYCLE, COST-ESTIMATING MODEL FROM COBOL TO FORTRAN IV

1. INTRODUCTION

The Cost and Analysis Division of the U.S. Army Missile Command (MICOM) developed a Time-Phased, Life-Cycle, Cost-Estimating Model which has gained wide acceptance throughout the Army Materiel Command (AMC) and its subordinate commands for assessing the life cycle costs of Army Systems (Reference 1). The model is especially useful in that the input format corresponds with that described in AMC's Project ICE (Improved Cost Estimating) Phase III.

The program was originally written in the COBOL computer language. Because of a need for this model in the Systems Methodology and Resource Studies Office (SM&RSO) of the U.S. Army Materiel Systems Analysis Agency (USAMSAA) and the lack of COBOL capability on the Aberdeen Research and Development Center's (ARDC) computer, the model was converted to FORTRAN IV.

This memorandum describes the FORTRAN IV version of MICOM's Time-Phased, Life-Cycle, Cost-Estimating Model.

There are many cost measures which can be associated with an Army weapon/support system. Examples are procurement costs, operating costs, life-cycle costs, etc. The life-cycle cost estimate, associated with each system, describes the cost of that system from its R&D status through its operational phase and to its retirement, and therefore provides the most comprehensive portrayal of the cost of the system.

Realizing that life-cycle costs were playing an important role in its acquisition process, the Army recognized the need to standardize the procedure for determining these life-cycle costs. This procedure was defined in the Improved Cost Estimating Study (ICE-III)(Reference 2).

2. IMPROVED COST ESTIMATING (ICE-III)

Army Regulation 37-18 "...establishes a set of cost categories and elements to be used by weapon/support system cost analysis activities... These categories and elements will be used for life-cycle

[1] U.S. Army Missile Command, Time Phased Life Cycle Cost Model, Cost Analysis Division, 30 June 1969.

[2] Letter, Office of AMC Comptroller, Program for Improved Cost Estimating (ICE) Phase 3, 14 May 1970.

analyses of existing and proposed weapon/support systems, to establish a uniform basis for cost analysis... and are to be used in life cycle costs analyses for cost effectiveness and decision oriented studies... The objective of this regulation is to improve cost... estimating procedures within the Department of Army through the use of uniform categories and elements for weapon/support system costs." (Reference 3).

AR 37-18 defines Cost Categories as: "The major divisions of weapon/support systems' cost from inception to retirement of the system." (Reference 3). The following cost categories are defined in detail in AR 37-18:

- Research and Development
- Investment Non-Recurring
- Investment Recurring
- Operating

Cost Elements are defined as "The subdivision of cost categories related to work areas or processes performed in developing, producing, and operating a weapon/support system." (Reference 3).

A complete breakdown of the cost structure associated with weapon/support systems is given later in this report. The cost structure proposed in AR 37-18 forms the basis for the rationale used in this cost estimating model

3. DESCRIPTION OF ICE-II I CODES AND INDEXING STRUCTURE

The chart on the following page describes the ICE-III codes and indexing structure. These are used to create codes for each level of data that are input to the life-cycle cost model. A summary of these codes is now given:

Pair of Digits	Represents
1	The system under consideration
2	Cost Category
3	Cost Element
4	Type of Cost

[3] Army Regulation AR 37-18, "Weapon/Support Systems Cost Categories and Elements," 2 July 1968.

ICE-PHASE III AND AR 37-18 CODES AND INDEXING STRUCTURE

1st Pair of Digits
Represent System

2nd and 3rd Pairs of Digits
Represent Cost Category and Element

01 Development
- 01 Engineering
- 02 Tooling
- 03 Prototype Prod.
- 04 System Test & Eval.
- 05 Data
- 06 Total Sys. Mgt.
- 07 Construction
- 08 Training

02 Inv. Non-Recur.
- 01 Adv. Prod. Eng.
- 02 Tooling
- 03 Prod. Base Supp.
- 04 Data
- 05 Non-Site Const.
- 06 Inst'r Tng-Crew
- 07 Inst'r Tng-Maint
- 08 Total Sys. Mgt.

03 Inv. Recur.
- 01 Engineering
- 02 Tooling
- 03 Quality Cont.
- 04 Manufac.
- 05 Purch. Equip.
- 06 Mat. Overhead
- 07 Data
- 08 Sub-contract
- 09 Init. Trng-Crew
- 10 Init. Trng-Maint.
- 11 Init. Prov-Spares
- 12 Init. Prov-Ammo.
- 13 1st Dest. Trans.
- 14 Modifications
- 15 Maint. of Prod. Base
- 16 Total Sys. Mgt.
- 17 Site Construction
- 18 Integration

04 Operating
- 01 Personnel-Crew
- 02 Personnel-Maint.
- 03 Cons-Non-Rep Parts
- 04 Cons-Rep Parts
- 05 Cons-Ammo.
- 06 Cons-Pol
- 07 Cons-Elec.
- 08 Integ Log. Sup.
- 09 Data
- 10 Equip. Trans.
- 11 2nd Dest. Trans.

4th Pair of Digits
Represent Type of Cost

- 01 Contract
- 02 In-House
- 03 Below Depot-C
- 04 Below Depot-In-House
- 05 Depot-Cont.
- 06 Depot-In-House

5th Pair of Digits
Represent Sub-Element

- 01 Direct Labor
- 02 Materials
- 03 Overhead
- 04 Other Direct Chrgs.
- 05 Gen. & Adm.
- 06 Profit
- 07 Pay & Allow
- 08 Repl Trng.

6th Pair of Digits
Represent Appropriation

- 01 RDT&E
- 02 PEMA
- 03 O&MA
- 04 MPA
- 05 MCA
- 06 ASF
- 07 FHMA

7th-8th-9th Pairs of Digits
Represent the Work Breakdown Structure
of the Particular System Under Study

5	Sub-Element
6	Appropriation
7-9	Work Breakdown Structure of System

The code associated with each level of data contains N pairs of digits where $N \leq 9$. For illustrative purposes we will now describe levels of data, and assign their associated level codes from the indexing structure.

Example 1: Suppose a level in which we are interested is the RDT&E costs of direct labor, of in-house engineering in the development cost category. The associated code would be as follows:

Code:	01	01	01	02	01	01
Pair No.:	1	2	3	4	5	6

Pair 1: System code

Pair 2: Development cost category

Pair 3: Engineering-cost element

Pair 4: Type of cost-in-house

Pair 5: Sub-element-direct labor

Pair 6: Appropriation-RDT&E costs

Example 2: The level code for the MPA costs of Direct Labor under contract for crew initial training is:

Code:	01	03	09	01	01	04
Pair No:	1	2	3	4	5	6

Pair 1: System code

Pair 2: Cost category-investment Recurring

Pair 3: Cost element-initial training crew

Pair 4: Type of cost-contract

Pair 5: Sub-element-direct labor

Pair 6: Appropriation-MPA

In this manner the codes are assigned to each individual level of data. Notice that no codes were given for the work breakdown structure, since it will be peculiar to each system under consideration.

This code and indexing structure allows a great deal of latitude in the amount of detail which is to be used in the model. This level of detail is to be defined by the individual analyst and will be dependent upon his time and resource constraints on data acquisition.

4. THE MODEL

The manner in which the life-cycle costs are derived in LICEM is easily understood once the cost breakdown is defined by ICE-III and AR 37-18. The data are stratified in as many as nine levels of complexity, with the greatest complexity occurring at the work breakdown structure level, and the least complexity occurring at the total systems cost level. Since a code is associated with each level of data, the final cost associated with a particular level is the sum of the costs associated with its sublevels. In this manner the data are "summed up" until the total system cost is determined.

The costs for each level of data are entered into the model through a set of input cards. There are seven distinct types of cards which could be used to calculate the cost associated with a level of data. The description of these cards, their use and format, will be presented in Section 9.1, Preparation of Input.

5. SUMMING UP PROCEDURE

As stated previously, due to the "summing up procedure," only nine levels of data are needed for storage in core during the operation of the processing model. See Figure 1 for visual description of storage matrix used in Main Processing Program. The nine levels in storage, one for each of the nine possible pairs of digits in the level code, represent the latest level of data entered, composed of 1,2,3,...,9 pairs of digits, respectively. As a new level of data (whose code is composed of $N(N\leq9)$ pairs of digits) is entered and its attendant costs calculated, they will replace the current data in the N^{th} row of the storage matrix. But before this transfer can take place, the "summing up" procedure must be performed. This procedure can best be explained by example, and one is given in this section. Before the example is given, however, a few words on how the "summing up" procedure works in general would be useful. One can think of all the sublevels of one level (super level) as being nested, or contained within the super level. The summing up procedure then totals the contributions of all nested sublevels, in turn giving the total for each sublevel in the set, as well as the total for the super level.

	Level NUM	Cost YR 1	Cost YR 2	Cost YR 3		Cost YR 29	Cost YR 30
Present Data of Degree 1							
Degree 2							
Degree 3							
Degree 4							
Degree 5							
Degree 6							
Degree 7							
Degree 8							
Degree 9							

Figure 1 Pictorial Description of Storage Matrix

These nine possible degrees* of data are all that need to be kept in memory at any one time. Once a level and its sublevels, if any, have been calculated and placed in the matrix, they are added upward and then written on an output tape, thus clearing those rows for the introduction of new data.

For illustration purposes, suppose our life-cycle cost is composed of five levels of data, covering a period of 2 years, that are in the first column of Figure 2. The times, noted along the left margin correspond to the entrance of the next level of data. The STORAGE MATRIX and OUTPUT TAPE columns are snap-shot views of how the matrix and tape appear during each stage of this processing example.

TIME=0: this level is used merely to show the levels of data that are to be entered, and that the storage matrix and tape are initially blank.

TIME=1: enter the first level of data. Note that these data go directly to the storage matrix. Also, nothing will be placed on the

*The degree of a level of data is the number of pairs of digits present in its level number, i.e., 0102030501 is of degree 5.

	Level Data To Be Entered				Storage Matrix Code	Costs		Output Tape Code	Costs	
	01	10	15	LEV1						
	0101	8	9	LEV2	"Blank"			"Blank"		
TIME=0	010101	5	6	LEV3						
	010102	4	8							
	0102	10	8							
	Enter 1st			LEV1	01	10	15			
TIME=1	Level of Data			LEV2	"Blank"			"Blank"		
				LEV3	"Blank"					
	Enter 2nd			LEV1	01	10	15			
TIME=2	Level of Data			LEV2	0101	8	9	"Blank"		
				LEV3	"Blank"					
	Enter 3rd			LEV1	01	10	15			
TIME=3	Level of Data			LEV2	0101	8	9	"Blank"		
				LEV3	010101	5	6			
	Enter 4th			LEV1	01	10	15	010101	5	6
TIME=4	Level of Data			LEV2	0101	13	15			
				LEV3	010102	4	8			
	Enter 5th			LEV1	01	27	38	010101	5	6
TIME=5	Level of Data			LEV2	0102	10	8	010102	4	8
				LEV3	"Blank"			0101	17	23

At this time all data have been entered,

The final summation procedure yields

	010101	5	6
Final Form of	010102	4	8
Output Tape	0101	17	23
	0102	10	8
	01	37	46

Figure 2 Level Data, Storage Matrix, and Output Tape For Example of Summing Up Procedure

output tape until one level replaces another in the storage matrix, and the "summing up" procedure is performed.

TIME=2: enter the second level of data. This level code 0101, composed of two pairs of digits is placed directly into the second row of the storage matrix.

TIME=3: enter the third level of data. This is placed in the third row of the storage matrix. At this point no "summing up" has been done.

TIME=4: enter the fourth level of data, 010102; note that this replaces 010101 in the storage matrix; hence 010101 must be added up and written on the output tape. The adding up is evident from the values in 0101 in the storage matrix, which are now 13, 15. These values are the sum of 0101's original values 8 and 9 and the values 5 and 6 from 010101. Finally 010101 and its original values (5,6) are written on tape.

TIME=5: enter the fifth level of data, 0102, which will replace 0101 in the storage matrix. Several things need be done here.

- All sublevels of 0101, (which is 010102) must be "summed up" and included in 0101. Note in the output tape that this yields a final total for 0101 of 17 and 23.

- After being summed up, these sublevels of 0101 must also be written on the output tape. Note that level 010102 with costs 4 and 8 appear on the image of the output tape.

- Before 0101 is written on tape, its final values must be summed to level 01, given 01 values of 27 and 38 in the storage matrix.

- Finally 0102 replaces 0101 in the second level of the storage matrix.

TIME-FINAL: At this time all levels have been entered. "Summing up" the costs of 0102 to 01, giving totals of 37 and 46, and writing these final two levels onto the output tape completes the process.

The coding structure of ICE-III makes possible the use of this "summing up" procedure. This procedure permits the life-cycle cost estimate to include as many levels as desired, since regardless of the length of the input only nine levels of data need to be stored in memory at any one time.

6. DISCUSSION OF THREE PROGRAM SEQUENCE

A sequence of three programs is used to determine the time-phased life-cycle cost estimates. The three programs are:

- Data check and Input Tape Preparation Program;
- Main Processing Program; and
- The Output Preparation and Summarization Program.

These programs and their functions are described below in more detail.

6.1 Data Check and Input Tape Preparation Program.

The functions of the first program are:

- To prepare a tape of data which will be used as input to the Main Processing Program;
- To give a print-out of data by card type, which will facilitate the location of format errors;
- To check for discontinuities in the data. A discontinuity is a difference of two or more degree levels between any level and its initial sublevel (e.g., 01 and 010101 with no 4-digit code such as 0101 in between).

It is important to note here that the "summing up" procedure, which is used in the main processing program, will work correctly only if the level data are ordered in an increasing degree of complexity. The function of ordering the level data, according to the above rule, has not been included in this program, due to excessive sorting time for large samples and the fact that the data need not be sorted for each new run. Thus, an optional program is provided which will order the input data and provide a tape of the ordered data for input to the main processing program.

Listing and flow charts of this optional program are given in the appendix. The input for this program assumes that the data are in the following order:

- The three "L" cards, in order;
- The function cards, in any order;
- A card with 10 asterisks following the function cards;

- The level data, in any order; finally
- Two blank cards and a PROB card.*

6.2 Main Processing Program.

This program is the heart of the cost-estimating model; its functions are:

- To read the tape prepared by Input Tape Preparation Program, or the optional program;
- To process these data and obtain the life-cycle cost estimates of the weapon/support system by level and time interval;
- To write an output tape containing the results of the above levels, which will be used as input to the Report Preparation and Summarization Program.

6.3 Report Preparation and Summarization Program.

This is the final program in the sequence; its functions are:

- To read as input, the tape prepared by the Main Processing Program;
- To sort this tape, by level, into order of increasing complexity of level number;
- To give standard output of the calculated costs by level number;
- To provide an option for obtaining three standard summaries of the cost data by:
  - major cost category;
  - appropriation; and
  - cost category by appropriation.

It should be noted here that the three program sequence will function correctly as long as the input cards are in correct form even though specific level numbers are not associated with ICE-III. However,

*If the level data are not correctly ordered the optional sorting program must be used.

if this is the case, the three standard summaries, described above, will not preform correctly, since in the ICE format cost categories and appropriations have been given specific codes.

Figure 3, gives a flow diagram of the three program sequence, and the necessary steps in obtaining a life cycle cost estimate.

7. DESCRIPTION OF INPUT CARDS

There are nine card types which may be used as input to the life-cycle cost-estimating model. Card types A, B, C, D, E, G, and H are used to associate data with a specific level of data or group of levels. The L cards carry information pertinent to the introduction of the entire study. The F cards enter function data, which can be used by any data level or group of levels.

7.1 Summarization of Card Types and Their Uses.

- "A" used to associate a name with each specific level number. A card type "A" must be present with each distinct level number. Two cards can be used.*

- "B" used to enter cost or quantity data into the data tables. Each set can contain as many as five cards depending upon which of three input forms is used.**

- "C" used to reference "functions" which are to be associated with the specific level number. From one to five function tables can be referenced on each card, and as many as nine "C" cards can be used at each level.

- "D" used to enter data into a special function table FN98. The data are stored here until a new set of "D" cards are encountered. Each set contains from one to five cards depending upon which of the three formats are used. (Later references on "C" card to FN98 will recall the values previously defined in "D" cards).

*The second "A" card would be a continuation of the first "A" card.

**There are three distinct forms of "B" and "D" cards. The forms may be combined when defining data for a specific data level.

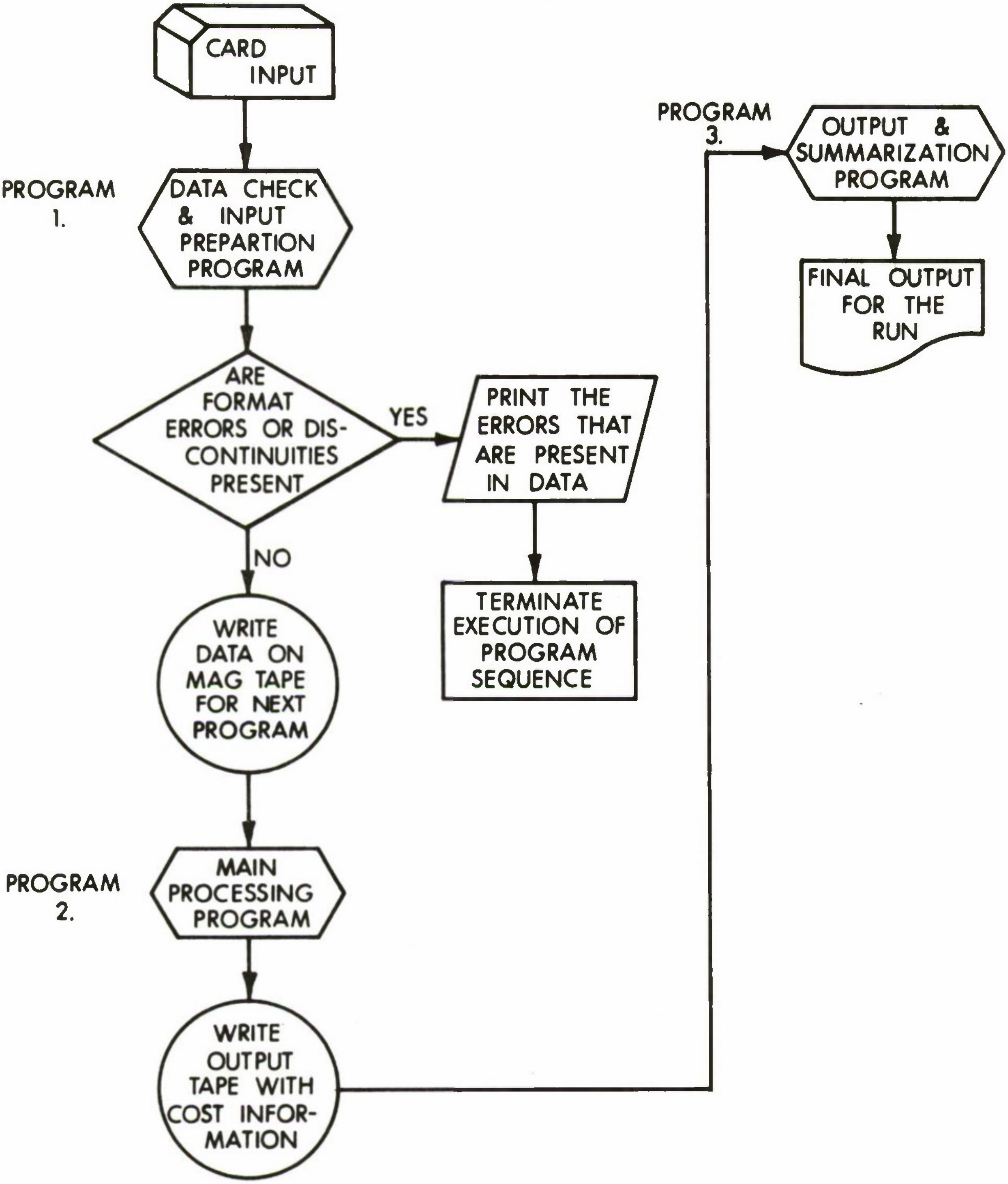

CARD INPUT
PROGRAM 1.
DATA CHECK & INPUT PREPARTION PROGRAM
ARE FORMAT ERRORS OR DIS-CONTINUITIES PRESENT
YES
PRINT THE ERRORS THAT ARE PRESENT IN DATA
TERMINATE EXECUTION OF PROGRAM SEQUENCE
NO
WRITE DATA ON MAG TAPE FOR NEXT PROGRAM
PROGRAM 2.
MAIN PROCESSING PROGRAM
WRITE OUTPUT TAPE WITH COST INFOR-MATION
PROGRAM 3.
OUTPUT & SUMMARIZATION PROGRAM
FINAL OUTPUT FOR THE RUN

Figure 3 Pictoral Description of 3 Program Sequence.

- "E" used to form a new special function, FN99, through the addition of one through five separate function tables. These "E" cards should be used only when the derived function table is related to only one specific level number. Data are retained in this table until another set of "E" cards is encountered. (Later references to FN99 on "C" card will recall values defined in "E" cards).

- "F" used to generate from 1 to 97 different function tables. Each set contains from one to five cards depending upon which format is used. "F" cards are not read or associated with any specific level number, and can be referenced through "C" cards from any level.

- "G&H" must be used together. "G" cards contain cost data and "H" cards contain quantity data. "G" cards are used to generate cost data based upon learning curve calculations.

- "L" used as report header and beginning and ending year of the study. There are three distinct "L" cards used with each run, and are located at the very beginning of the data deck.

The general format of data cards A, B, C, D, E, and G is:

Card Columns	Variable	Format	Variable Description
1-18	Level 1(I) I=1,8	18I1	The level number associated with this particular set of data
19	CARDNO	I1	The number of the card which is associated with this type for this level number
20	Blank	1X	
21	Card	A1	The letter designating the card type
22-80	----	Variable	These columns contain the data, in different formats for distinct card types, which are to be used in the calculation of this level cost

7.2 Description of Input Cards by Type.

As was stated previously columns 1 through 21 contain the same data for card types A, B, C, D, E, G, and H. These cards vary only in Columns 22-80, and so the following description of the card types will be concerned only with these columns.

"A" Data Card Format: Used to associate a name or description with each specific level of data. At least one "A" card must be used with each level, while up to two are allowed.

Card Columns	Variable	Format	Description of Variable
22-80	INAME1(I), I=1,6	A9,5A10	Contains the name of this data level

If two "A" cards are used the only difference would appear in Column 19, where a 2 would be placed to represent that this was the second "A" card used for this level, and Columns 22-80 would contain INAME1(I), I=7,12.

"B" Data Card Format: Used to enter data into tables which will be used in the calculation of the cost pertinent to this level of data. The three types of "B" cards discussed below, enter the data into an array, where each member is associated with a given interval during the time frame of the study.

B-TYPE I: If the data for this level are constant for each time increment over the period of the study, the B-TYPE I card should be used.

Card Columns	Variable	Format	Description of Variable
22-24	Constant	A3	The word "ALL"
25	Blank	1X	
26-32	Value	F7.3	The value here will be placed in the BDATA array for all time increments during the study

B-TYPE II: If the data to be entered through the "B" card are constant over some period during time frame of the study, then a B-TYPE II card would be used to enter the data for this period.

Card Columns	Variable	Format	Description of Variable
22-25	Constant	A4	The word "FROM"
26	Blank	1X	
27-28	IYR1	I2	Initial year of period during which the data are constant
29	Blank	1X	
30-31	Constant	A2	The word "TO"
32	Blank	1X	
33-34	IYR2	I2	Final year of period during which the data are constant
35	Blank	1X	
36-42	Value	F7.3	Value which is to be entered into the data tables from IYR1 to IYR2

B-TYPE III: If the data to be entered with "B" cards vary from year to year, then TYPE III would be used.

Card Columns	Variable	Format	Description of Variable
22-23	IYR(1)	I2	Period during study which will have the value given in the next field
24-30	Value(1)	F7.3	Value to be placed in B table for IYR(1)
31-32	IYR(2)	I2	Same definitions apply for IYR(I), Value(I), I=1,6
33-39	Value(2)	F7.3	
40-41	IYR(3)	I2	
42-48	Value(3)	F7.3	
49-50	IYR(4)	I2	
51-57	Value(4)	F7.3	

58-59	IYR(5)	I2
60-66	Value(5)	F7.3
67-68	IYR(6)	I2
69-75	Value(6)	F7.3
76-80	Blank	I2

"C" Data Card Format: Used to reference functions containing factors to be used in the calculation of cost data for this level. From one to five functions may be referenced on each "C" card and a maximum of nine "C" cards may be used with any one level.

Card Columns	Variable	Format	Description of Variable
22-23	Constant	A2	The letters "FN"
24-25	IFN(1)	I2	Number of the 1st function which is referenced
26-27	Constant	A2	The letters "FN"
28-29	IFN(2)	I2	Number of the 2nd function which is referenced
30-31	Constant	A2	The letters "FN"
32-33	IFN(3)	I2	Number of the 3rd function which is referenced
34-35	Constant	A2	The letters "FN"
36-37	IFN(4)	I2	Number of the 4th function which is referenced
38-39	Constant	A2	The letters "FN"
40-41	IFN(5)	I2	Number of the 5th function which is referenced

"D" Data Cards: Used to enter either cost or quantity data, used in the calculation of costs pertaining to a specific level. There are three types of "D" cards, and they have exactly the same format and interpretation as the three types of "B" cards.* Therefore, they will not be given here. "D" cards also have another particular use. The data

*With the obvious exception that a "D" would appear in Column 21.

entered by these cards are placed in special function 98, and will be stored there until another set of "D" cards is encountered. Hence, these data can be used many times subsequent to their appearance on "D" cards by referencing function 98 on either "C" or "E" cards.

"E" Data Cards: Have a use which is similar to that of "C" cards. Recall that "C" cards reference functions which are used as factors in the calculation of a particular level's cost data. "E" cards also reference functions. The functions referenced are added by year or time increment and in essence form a new function. This new function is then used as a factor in the calculation of costs for the level.

Card Columns	Variable	Format	Description of Variable
22-23	Constant	A2	The letters "FN"
24-25	IFN(1)	I2	Number of 1st function referenced
26-27	Constant	A2	The letters "FN"
28-29	IFN(2)	I2	Number of 2nd function referenced
30-31	Constant	A2	The letters "FN"
32-33	IFN(3)	I2	Number of 3rd function referenced
34-35	Constant	A2	The letters "FN"
36-37	IFN(4)	I2	Number of 4th function referenced
38-39	Constant	A2	The letters "FN"
40-41	IFN(5)	I2	Number of 5th function referenced

"E" data calculated in this manner are stored in special function 99 until a new set of "E" cards are encountered. Thus, the data can be entered once then recalled by referencing function 99 on a "C" card.

"F" Data Cards Format: Used to enter data into the function Tables 1 through 97. (Recall that "D" and "E" cards enter data into functions 98 and 99 respectively. Data input on "F" cards should not be placed in either special function 98 or 99.) The data entered into the function tables are not associated with a specific level but can be used as factors in calculating the costs pertaining to any level.

Three types of "F" cards are used to enter data into function Tables 1 through 97. The uses of the three types of cards are the same as those for "B" and "D" cards.

TYPE 1 is used when all values in the function are to be a single constant.

Card Columns	Variable	Format	Description of Variable
1-14	Blank	14X	
15-16	Constant	A2	The letters "FN"
17-18	Function Number	I2	Number of function into which we are placing data
19	CARDNO	I1	Number of card pertaining to this function
20	Blank	1X	
21	Constant	A1	The letter "F"
22-24	Constant	A3	The word "ALL"
25	Blank	1X	
26-32	Value	F7.3	Value to be placed in all entries of this function

TYPE 2 is used when the value in the function remains constant over some interval of years, less than the entire period of the study. Columns 1-21 are the same as Type 1.

Card Columns	Variable	Format	Description of Variable
22-25	Constant	A4	The word "FROM"
26	Blank	1X	
27-28	IYR1	I2	Beginning FY of period
29	Blank	1X	
30-31	Constant	A2	The word "TO"
32	Blank	1X	
33-34	IYR2	I2	Final FY of period
35	Blank	1X	
36-42	Value	F7.3	Value to be placed in the function table for this period
43-80	Blank	38X	

TYPE 3 is used when the values to be placed in the function table vary from year to year. Columns 1-21 are the same as Type 1.

Card Columns	Variable	Format	Description of Variable
22-23	IYR(1)	I2	Period during study for which this function's value will be specified in next field
24-30	VAL(1)	F7.3	Value given this function for period given in preceding field
31-32	IYR(2)	I2	Definitions same as above
33-39	VAL(2)	F7.3	
40-41	IYR(3)	I2	
42-48	VAL(3)	F7.3	
49-50	IYR(4)	I2	
51-57	VAL(4)	F7.3	
58-59	IYR(5)	I2	
60-66	VAL(5)	F7.3	
67-68	IYR(6)	I2	
69-75	VAL(6)	F7.3	

"G&H" Data Card Format: Used to generate cost and quantity data for a specific level. An "H" card must be used each time a "G" card is used. The "G" card contains the cost data, and references a function which contains the quantity to be procured or developed. The "H" card contains further specifications concerning quantities. These variables are then used in "learning curve" calculations to determine the costs for this level during each time increment. The uses of the "G" and "H" cards will be explained in more detail in a later section.

Card Columns	Variable	Format	Description of Variable
22-30	A	F9.0	First unit cost for the item described in this level

31-36	B	F6.6	The value of B in learning curve formula
37-38*	BPR	I2	Slope, which is equal to $- \log B/_{-\log 2}$
39-44	IQTY1	I6	Starting quantity for the calculations
45-46	FN	A2	Constant
47-48	NUM	I2	The number of the function which contains the quantity data pertinent to this level

"H" Card Format

Card Columns	Variable	Format	Description of Variable
22-25	UNIT(1)	I4	The number of the unit which describes the beginning of the first quantity category
26-31	QTY(1)	F6.3	The factor to be multiplied by each member of the above referenced function, which lies between UNIT(1) and UNIT(2)
32-35	UNIT(2)	I4	The number of the unit which describes the end of the first quantity category and the beginning of the second quantity category
36-41	QTY(2)	F6.3	The factor to be multiplied by each member of the above referenced function, which lies between UNIT(2) and UNIT(3)
42-45	UNIT(3)	I4	Same definition as previous unit variables

*This entry is superfluous but was included due to its presence in MICOMS card description.

46-51	QTY(3)	F6.3	Same definition as previous QTY variables
52-55	UNIT(4)	I4	Same definition as previous UNIT variables
56-61	QTY(4)	F6.3	Same definition as previous QTY variables
62-65	UNIT(5)	I4	Same definition as previous UNIT variables
66-71	QTY(5)	F6.3	Same definition as previous QTY variables

"L" Data Card Format: Contains header information and the beginning and ending fiscal years of the study. There are three cards, and all must be present for each model run.

L-1:

Card Columns	Variable	Format	Description of Variable
1-16	Blank	16X	
17-18	Constant	A2	The letters "LA"
19	Constant	I1	The number "1"
20	Blank	1X	
21	Constant	A1	The letter "L"
22-80	Report Name	A59	

L-2:

1-16	Blank	16X	
17-18	Constant	A2	The letters "LA"
19	Constant	I1	The number "2"
20	Blank	1X	
21	Constant	A1	The letter "L"
22-50	Report Name Continued	A29	

51-52	Day	I2	Day of Run
53	Blank	1X	
54-62	Month	A9	Month of Run
63	Blank	1X	
64-67	Year	I4	Year of Run
68-80	Blank	12X	

L-3:

1-16	Blank	16X	
17-18	Constant	A2	The letters "LA"
19	Constant	I1	The number "3"
20	Blank	1X	
21	Constant	A1	The letter "L"
22-23	IYR1	I2	Beginning Fiscal Year of Study
24-25	Blank	2X	
26-27	IYR2	I2	Ending Fiscal Year of Study

7.3 Further Explanation of "G" and "H" Cards.

The "G" and "H" cards, as stated previously, are used to determine the costs associated with a level of data through learning curve calculations. Some additional explanation is necessary here to make clear the use of these card types.

The learning curve function has the form:

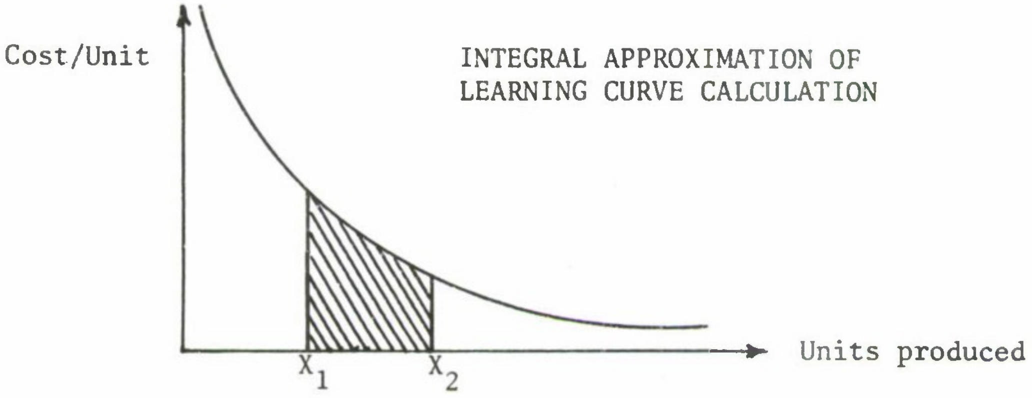
$$Y = AX^{-B}$$

A = first unit cost of the item being produced

B = slope of learning curve

X = quantity being produced

Y = cost of the Xth item



The total cost of units X_1 through X_2 would be

$$\text{Total Cost} = \sum_{X=X_1}^{X_2} AX^{-B} \quad .$$

However, for large numbers of items ($N \geq 30$), LICEM uses the integral approximation:

$$\text{Total Cost} = \int_{X_1-1}^{X_2} AX^{-B}$$

$$= \frac{A}{1 - B} \left\{X^{1-B}\right\}_{X_1-1}^{X_2} \quad .$$

The function referenced in Columns 47 and 48 of the "G" card contains the procurement quantities for the time increments during the study. These quantities coupled with the quantity to start (Columns 39-44) are sufficient to calculate the costs associated with each time period.

If one is calculating costs associated with the unit quantities referenced in the function in Columns 47 and 48, the "H" card must still be used, but it is really of little value. For example, suppose that function 5 is referenced, and contains the number of airframes to be procured during the time increments of the study, the "H" card for this example would then be:

COL 22-25	0001
COL 26-31	000001

This can be interpreted as the procurement of one airframe for each total missile to be procured.

Suppose now that in another level the costs of the fins to be placed upon each missile are to be calculated and that there are to be four fins/missile. In addition, five extra fins are required for testing. Function 5 would still be referenced in this level, since this function contains the number of missiles to be procured during each period. (The other entries on the "G" card would be changed to reflect the different learning curve parameters associated with the production of missile fins.) The "H" card in this example would be:

COL 22-25	0001	Units 1-5 have 5 fins/missile
COL 26-31	000005	
COL 32-35	0006	Units of more than 5 have 4 fins/missile
COL 36-41	000004	

These data have the following interpretation:

Associated with units 1-5 of the missiles referenced in function 5, we are using five fins/missile. (These five extra fins are those desired for testing.) Missile units above 5, have four fins/missile associated with them.

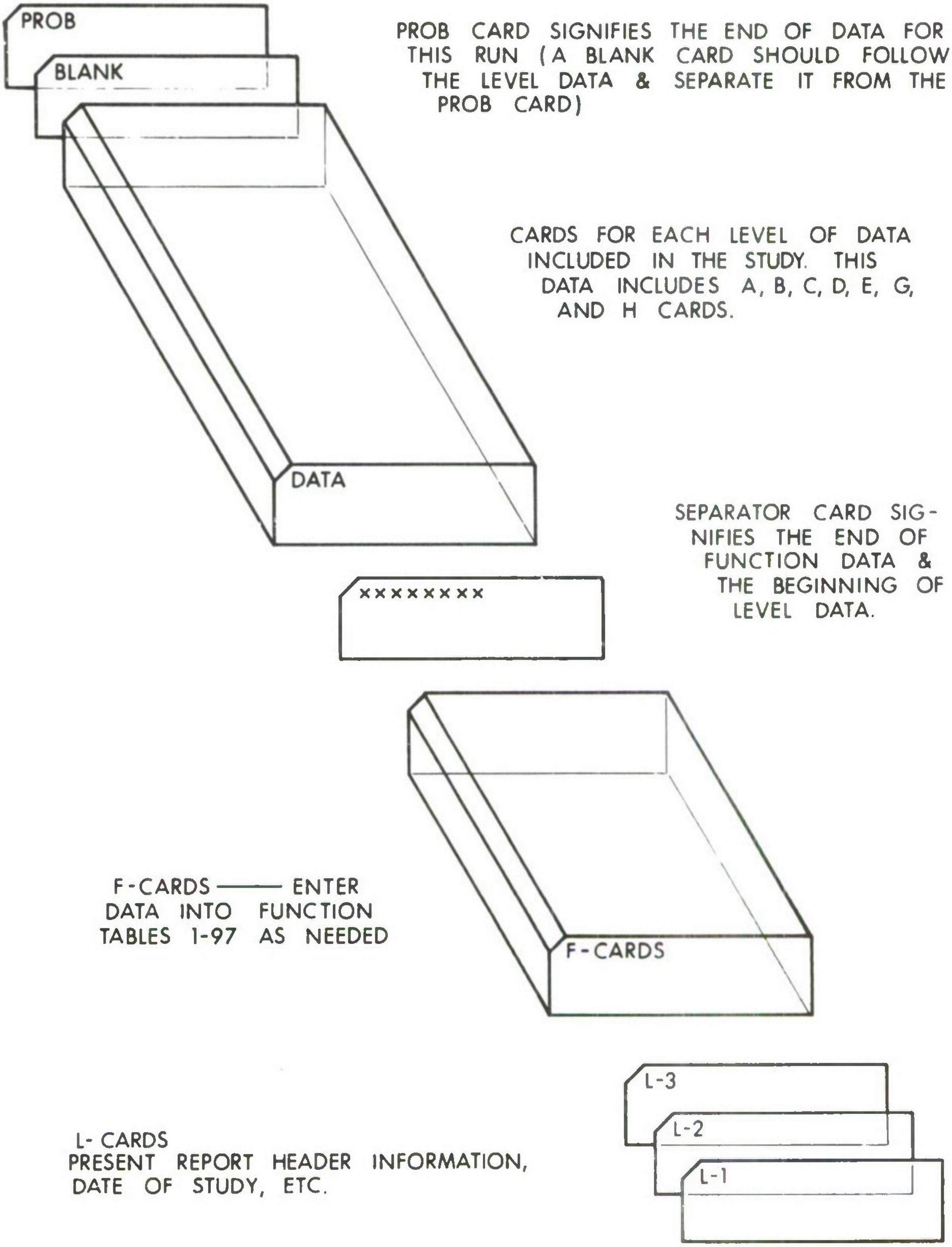
The use of the "H" card is now obvious, because through its use the quantities of each part of the work breakdown structure need not be stored in the function tables. As long as the number of each component used in a single unit is known, only the number of complete units needed must be stored in the function tables. This greatly reduces the amount of data which needs to be stored in core during a case.

Having described the cards and their functions we are now ready to describe the order of input cards for this model. Figure 4 pictorially describes the necessary order of input. Note the order:

1st - 3 "L" cards

2nd - All function card data

3rd - Card with 10*'s in 1st 10 columns (used to separate function cards from level data)



PROB CARD SIGNIFIES THE END OF DATA FOR THIS RUN (A BLANK CARD SHOULD FOLLOW THE LEVEL DATA & SEPARATE IT FROM THE PROB CARD)

CARDS FOR EACH LEVEL OF DATA INCLUDED IN THE STUDY. THIS DATA INCLUDES A, B, C, D, E, G, AND H CARDS.

SEPARATOR CARD SIGNIFIES THE END OF FUNCTION DATA & THE BEGINNING OF LEVEL DATA.

F-CARDS —— ENTER DATA INTO FUNCTION TABLES 1-97 AS NEEDED

L- CARDS
PRESENT REPORT HEADER INFORMATION, DATE OF STUDY, ETC.

Figure 4 Order of Input Cards

4th - Level data, where all levels are ordered in increasing level complexity.

5th - Blank card

6th - Prob Card (signifies end of problem)

8. MISCELLANEOUS POINTS

Several additional points, not logically falling within the classification of any of the other sections, should be discussed. These points, to be discussed in the following sections, are:

- Ease of parameter change for sensitivity analysis
- Omission of original data sort
- Special use of "A" and "B" cards
- "B-D-F" card options
- Use of previously made decks for MICOM's COBOL version
- Summary of card combinations and their uses
- Tape requirements for three program sequence
- Description of tapes produced by the input tape preparation and the main processing programs
- Compilation and running times for the three program sequence

8.1 Ease of Parameter Change for Sensitivity Analysis.

The formats in which the data are entered into this life-cycle cost model are especially suited to the performance of sensitivity analyses. If the parameter to be investigated is stored in the function tables, a change of at most five cards is all that is necessary to change the variable value.

Also, if the variable of interest lies in only one portion of the life-cycle cost estimate, say the operating cost category, it is not necessary to recompute the costs for the other cost categories. This can prove to be a time and money saving feature on extensive studies.

8.2 Omission of Original Data Sort.

In the original COBOL version of this LCCE model, tasks are sorted into three programs. In the initial program, similar to our Input Tape Preparation Program, a sorting routine ensures that the data are sorted and input in correct order.

In this FORTRAN version of the model, users are provided with two options. These are:

a. Data Check and Tape Input Preparation Program assumes that the data are in correct order with three main functions;

- to prepare a tape containing input for LCCE model.
- to list data, by card type, so that a visual check can be made for format errors.
- to check for discontinuities in the data.

b. Sorter assumes that the data deck is out of order; this program has two main functions;

- to put the data into correct order for input to LCCE model.
- to prepare a tape for input to LCCE model.

These options eliminate the need for a complete sorting each time a deck is run. The sorting procedure is time consuming and unnecessary when a data deck is to be run more than once.

8.3 Special Use of "A" and "B" Cards.

It may be that the costs associated with some level of data may already be known and no calculation is necessary to determine them. In this case there are only two card types necessary for the cost description of this level. These are:

- "A" - Card(s): to describe the level.
- "B" - Card(s): to enter costs for this level.

When this configuration of cards is used to describe a level of data, the model assumes that the costs entered on the "B" cards are entered in millions, even though the same format (F7.3) is used to enter the number.

As an example, if a "B" card was to be used to enter already known costs, then $1,500,000.00 would be entered as bbb1500, the decimal point would be assumed to lie between the 1 and 5.

8.4 "B-D-F" Cards (Options).

Note that there are three ways that data can be entered through the "B," "D," and "F" cards. These ways can be classified as "ALL" cards, "INTERVAL" cards, and "YEAR by YEAR" cards. "ALL" cards place the specified value into each year of the period. "INTERVAL" cards place the specified value into each year between and including the specified end-points, and the "YEAR by YEAR" cards specify a new value for each year

The order in which these cards appear is not commutative. That is, the same cards in different orders will not give the same results, as is seen in the examples below. (The example uses "F" cards, but the same caution applies to "D" and "B" cards.)

Example 1*

```
FN011 FALL-1000

FN012 F732000
```

This combination will have the following results:

70	71	72	73	74	75
1000	1000	1000	2000	1000	1000

If the two cards were reversed however, the results would be:

70	71	72	73	74	75
1000	1000	1000	1000	1000	1000

This result caused by the fact that the "ALL" card overlays its value on top of the 2000 entered by the previous card.

Consequently, there are many ways of entering a data set into storage correctly, but one must keep in mind that later cards overlay their values on those previously defined.

8.5 Use of Previously Made Data Decks for MICOM's Model.

Data decks which have been prepared for use in MICOM's life-cycle cost model, although written in COBOL, can be used as input to LICEM with the addition of one card in the data deck. This additional card contains 10 asterisks in the first 10 columns. It is inserted to separate function cards from level data.

*Assume time frame is 1970-1975.

8.6 Summary of Card Combinations and Their Uses.

In the following discussion, there is always an "A" card associated with the level under consideration:

Cards Present	Resulting Computations
"B" only	No computations made; assumed values on "B" cards are expressed in MILLIONS of dollars.
"B&C" or "C&D"	Multiplies those values in the "B" or "D" cards by the values present in the function(s) referenced on the "C" cards.
"B&C&D"	Multiplies the values on "B" cards by function(s) referenced on "C" card, and multiplies that product by values present on "D" card.
"B&E" or "D&E"	Multiplies the values in "B" or "D" cards by the sum of the functions (≥2) referenced on the "E" card.
"B&D&E"	Multiplies the values in the "B" and "D" cards and multiplies this product by the sum of the functions referenced on the "E" cards.
"G&H"	Used to make standard learning curve calculations of form $$C = \sum_{X=X_1}^{X_1} AX^{-B}$$
"G&H&B" or "C" or "D" or "E"	Makes standard learning curve calculations and multiplies results by data in "B" or "D" or by functions referenced in "C" or by the sum of the functions referenced in "E."

This list is not meant to be exhaustive but it does represent the basic card combinations used for obtaining costs in a given level.

Two Binary Coded Decimal tapes (BCD) are needed for the operation of the three program sequence. One tape is mounted on tape unit 1 and the other on tape unit 2. Following is a schematic diagram of how these tapes are utilized.

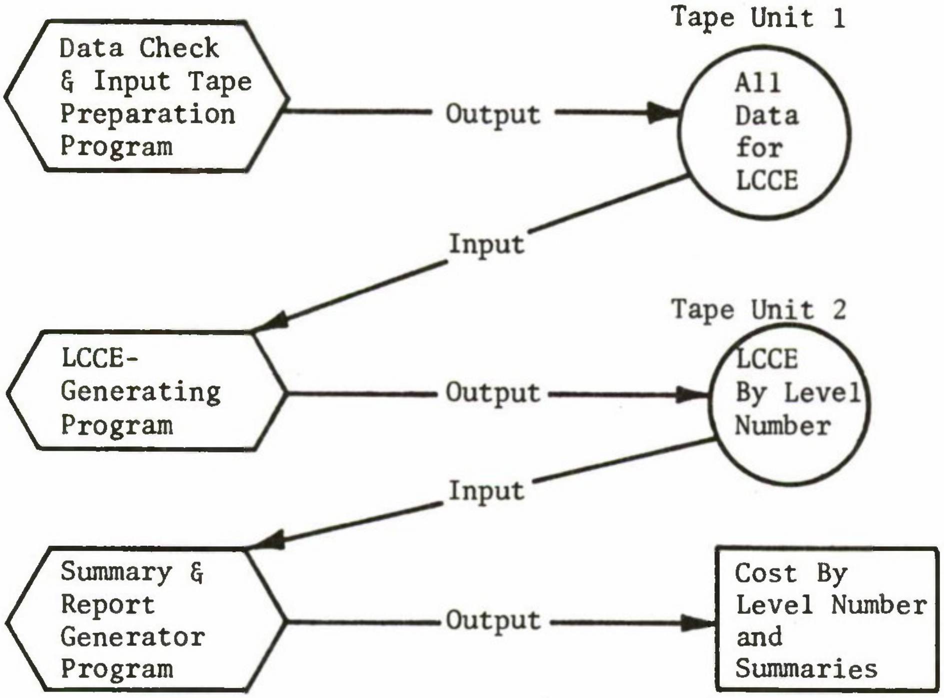


Figure 5 Tape Requirements for Three Program Sequence

8.7 Description of Tapes Produced.

1. The tape produced by the Input Tape preparation program consists of 80 column images of the card deck input. The arrangement and format is as follows:

	Format
L 1	8A10
L 2 Cards	8A10
L 3	8A10
Block of Function Cards	8A10
10*'s & 70 Blanks	A10, 70X
Level Data Including All Card Types	8A10
80 Blanks Marks End of Data	This tape is produced on unit 1.

2. The tape produced by the Main Processing Program is in the following form:

	Format
IYR1, IYR2	I4, 2X, I4
(REPNAM(I),I=1,9) NDAY, MONTH, NYR	A9, 5A10, A9, 2A10, 5X I2, 2X, A9, I4
ISTORE(J,K),K=1,2 (LEVEL NUMBER) ISTORE(J,K),K=4,12 IDENT. OF LEVEL)	A10, A8, A9, 5A10, A9, 2A10
STORE(J,K),K=1,30 COSTS OF JTH LEVEL)	(10F12.3/10F12.3/10F12.3)
LAST TWO RECORDS ARE REPEATED FOR EACH LEVEL OF DATA	
BLANK RECORD	
END FILE	

8.8 Compilation Time.

The compilation and running time for the three program sequence are given below: (run times are for example problem).

	Compilation Time (min)	Run Time (min)	Storage Requirements
Sorter	.47	.20	12K
LCCE	.90	1.03	10K
Report Generator	.47	.21	12K

Having all the concepts necessary to use the model, we are now ready to work through an example from input preparation to a description of the final output.

9. EXAMPLE PROBLEM

In this section, the time-phased, life-cycle cost estimates for the fictional X-4210 vehicle will be derived.

The FORTRAN coding sheets used to obtain the necessary input cards to LICEM are shown in Figures 6, 7, and 8. Note again the order of cards, "L" cards followed directly by the "F" (function) cards, a card with 10 asterisks, data by level (sorted into order of increasing complexity), a blank card, and finally a PROB card, which signifies the end of data deck.

First, note in our data set that each level begins its data with an "A" card, and that some levels have only "A" cards. When this is the case, the particular level, having only "A" card identification, will be the sum of the costs associated with all its sub-levels.

The normal type of calculation that will be required by the data will involve "B," "C," and "D" cards. When combinations of "B" and "C" or "C" and "D" cards are used, the data (cost or quantity) present on the "B" (or "D") card is multiplied by the data stored in the functions referenced by the "C" cards. Note that it is permissible to reference a function more than once in a "C" card. This has the effect of taking that function to a power. (See levels 010103010101 and 010103010102 for examples of these calculations).

Another commonly used calculation involves the "B" or "D" and "E" cards. In this case the data (cost or quantity) stored in the "B" or "D" card are multiplied by the sum of functions (by year) referenced in the "E" card. As in the "C" card, a function can be referenced more than once on an "E" card. This has the effect of multiplying the values in the function by the number of times it is referenced. (See level 010202 for example.)

The last standard calculation involves the "G" and "H" cards. These are used to perform learning curve calculations. (See level 010305 for example.) It is understood that the function referenced in the "G" card contains the quantities for the learning curve calculations.

This example can also help to make clear the definition of a discontinuity in the data. As defined earlier, a discontinuity in the data is a difference of two or more level degrees between a level and one of its initial sublevels. To paraphrase the definition, a level which contains N pairs of digits cannot be followed immediately by a sublevel containing N + 2 pairs of digits. This input error will cause the "summing up" procedure in the main processing program to operate incorrectly, and erroneous results will be given.

If discontinuities do exist in the data, SORTER will print out a statement indicating the level numbers between which the discontinuity exists.

9.1 Preparation of Input.

Notice in this example that the most detailed level of data contains only 12 digits. This level of detail was sufficient to illustrate the important functions of the program. The factors governing the level of detail for a user are amount of detail available and amount desired. Note that the last three pairs of digits are reserved for the work breakdown structure of the item being costed, and are not used in this example.

9.2 Output - Data Check and Input Tape Preparation Program (SORTER).

Having described the input for our example case we can now discuss the output as obtained from the three program sequence.

The objective of the print out from this sorter program is to help the user spot errors in his input data which could cause errors in the main processing model execution. Except for the "L" cards, all other card types are listed on separate pages at the top of which is an image of a correctly formated card, to help spot errors. Notice in cards "B" and "D" and "F" where numbers are to be entered three decimal places are assumed and no decimal point need be placed in the field. Figures 9 through 23 contain the output from program SORTER.

9.3 Output From Report Generator Program.

Note that the only output of the Main Processing Program is the tape it produces for the Report Generator and Summarization Program.

The output of the Report Generator and Summarization Program is divided into two portions:

a. (Listing of levels input and their associated costs over each period during the time frame of the study.) The levels are in the order in which they were input to the model. (Increasing order of Complexity). In addition to the cost by year, the total cost attributable to the level is printed on the right hand side of the output.

b. Optional Summaries (obtained by placing a 1 in column 1 of the first data card for the Report Generator and Summarization Program).

(1) Summary by Major Cost Category, where these cost categories include Development, Investment Recurring and Non Recurring, and Operating (Figures 24-25).

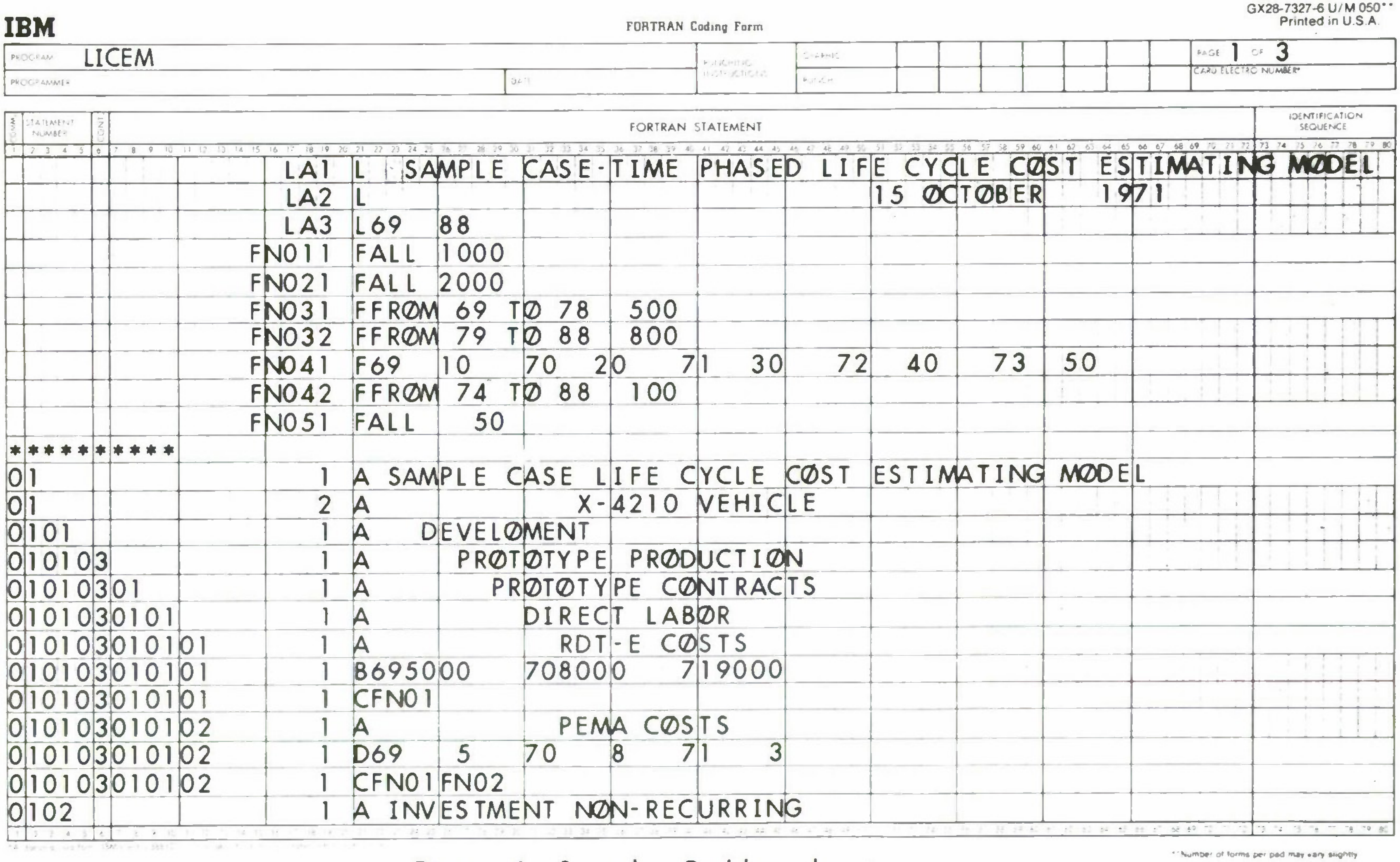
(2) Summary by Appropriation, where possible appropriations are RDT&E, PEMA, O&MA, MPA, MCA, ASF, FHMA. For this summary all levels, which are included in any appropriation, are summed, (regardless of the cost category in which they fall) to obtain the total for that appropriation. If any appropriation has a total of 0.0, no print out is given for that appropriation.

From our example the RDT&E present in the summary by appropriation is merely level 010103010101, since it is the only RDT&E level present.

The PEMA entry in this summary however is the sum of the costs of levels 010103010102 and 01040101010102. At the top of the page, under SUMMARY by APPROPRIATION, is given the total of the RDT&E and the PEMA costs are below (Figure 26).

(3) Finally the summary of all appropriations within each cost category is given (Figure 27).

It is possible that users might desire other summaries. For this reason the format of the output tape produced by the main processing program is given in the Miscellaneous Points Section. Once this format is known the extraction of any summary desired is relatively easy.



IBM FORTRAN Coding Form

GX28-7327-6 U/M 050**
Printed in U.S.A.

PROGRAM LICEM

PROGRAMMER

DATE

PUNCHING INSTRUCTIONS GRAPHIC PUNCH

PAGE 1 OF 3

CARD ELECTRO NUMBER*

STATEMENT NUMBER — CONT. — FORTRAN STATEMENT — IDENTIFICATION SEQUENCE

```
                LA1 L  SAMPLE CASE-TIME PHASED LIFE CYCLE CØST ESTIMATING MØDEL
                LA2 L                             15 ØCTØBER   1971
                LA3 L69  88
              FN011 FALL 1000
              FN021 FALL 2000
              FN031 FFRØM 69 TØ 78   500
              FN032 FFRØM 79 TØ 88   800
              FN041 F69  10   70 20    71  30   72  40   73  50
              FN042 FFRØM 74 TØ 88   100
              FN051 FALL   50
**********
01                1 A SAMPLE CASE LIFE CYCLE CØST ESTIMATING MØDEL
01                2 A            X-4210 VEHICLE
0101              1 A   DEVELØMENT
010103            1 A     PRØTØTYPE PRØDUCTIØN
01010301          1 A       PRØTØTYPE CØNTRACTS
0101030101        1 A         DIRECT LABØR
010103010101      1 A           RDT-E CØSTS
010103010101      1 B695000   708000   719000
010103010101      1 CFN01
010103010102      1 A           PEMA CØSTS
010103010102      1 D69   5   70   8   71   3
010103010102      1 CFN01FN02
0102              1 A INVESTMENT NØN-RECURRING
```

**Number of forms per pad may vary slightly

Figure 6 Sample Problem Input

IBM

FORTRAN Coding Form

GX28-7327-6 U/M 050**
Printed in U.S.A.

PROGRAM LICEM

PROGRAMMER

DATE

PUNCHING INSTRUCTIONS | GRAPHIC | PUNCH

PAGE 2 OF 3

CARD ELECTRO NUMBER*

COMM. | STATEMENT NUMBER | CONT. | FORTRAN STATEMENT | IDENTIFICATION SEQUENCE

```
010202            1 A    TØØLING
010202            1 BALL 1500
010202            1 EFN01FN02
0103              1 A   INVESTMENT RECURRING
010303            1 A     QUALITY  CØNTRØL
01030302          1 A      IN-HØUSE
0103030201        1 A         DIRECT LABØR
0103030201        1 BALL 9000
0103030201        1 CFN05
0103030202        1 A         MATERIALS
0103030202        1 BALL  500
0103030202        1 CFN01
010305            1 A       PRØCUREMENT CØSTS ØF VEHICLES
010305            1 G 9000.      .15        FNØ4
010305            1 H   1   2.0  25   1.0
0104              1 A ØPERATING CØSTS
010401            1 A   PERSØNNEL
01040101          1 A     IN-HØUSE
0104010101        1 A       DIRECT LABØR
0104010101        1 BALL   10
0104010101        1 CFN01FN03
010401010102      1 A         PEMA
010401010102      1 BFRØM 69 TØ 80   900
010401010102      2 BFRØM 81 TØ 89   800
```

*A standard card form, IBM electro 888157, is available for punching statements from this form

**Number of forms per pad may vary slightly

Figure 7 Sample Problem Input

IBM FORTRAN Coding Form

GX28-7327-6 U/M 050**
Printed in U.S.A.

PROGRAM LICEM

PROGRAMMER

DATE

PUNCHING INSTRUCTIONS — GRAPHIC / PUNCH

PAGE 3 OF 3

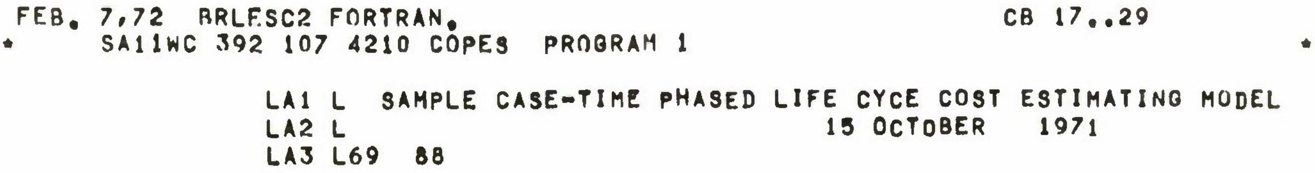
CARD ELECTRO NUMBER*

COMM.	STATEMENT NUMBER	CONT.	FORTRAN STATEMENT	IDENTIFICATION SEQUENCE

```
010401010102      1 CFN01

      PRØB


BELØW IS THE INPUT NECESSARY TØ ØBTAIN THE EXTRA SUMMARIES
      FRØM THE REPØRT GENERATØR PRØGRAM-

*     DATA
   1

      PRØB
```

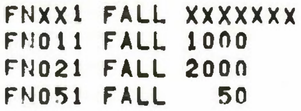
**Number of forms per pad may vary slightly

Figure 8 Sample Problem Input



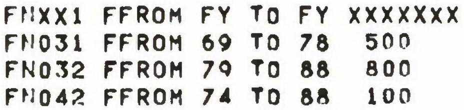
```
FEB. 7,72  BRLESC2 FORTRAN.                                   CB 17..29
*     SA11WC 392 107 4210 COPES  PROGRAM 1                                           *

             LA1 L  SAMPLE CASE-TIME PHASED LIFE CYCE COST ESTIMATING MODEL
             LA2 L                                 15 OCTOBER   1971
             LA3 L69  88
```

Figure 9   L-Cards From Sample



```
FNXX1 FALL XXXXXXX
FN011 FALL 1000
FN021 FALL 2000
FN051 FALL   50
```

Figure 10   F-All Cards From Sample



```
FNXX1 FFROM FY TO FY XXXXXXX
FN031 FFROM 69 TO 78  500
FN032 FFROM 79 TO 88  800
FN042 FFROM 74 TO 88  100
```

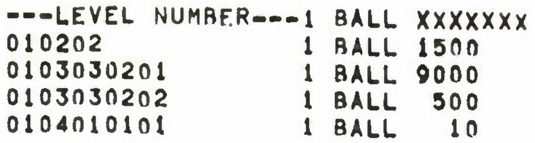
Figure 11   F-From Cards From Sample

```
FNXX1 FFYXXXXXXXFYXXXXXXXXYXXXXXXXFYXXXXXXXFYXXXXXXXFYXXXXXXX
FN041 F69   10     70   20     71   30     72   40     73   50
```

Figure 12 F-Year Cards From Sample

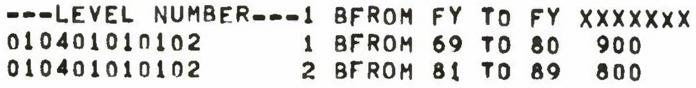
```
---LEVEL NUMBER---1 A-------------LEVEL IDENTIFICATION NAME------------
01                1 A SAMPLE CASE LIFE CYCLE COST ESTIMATING MODEL
01                2 A             X-4210 VEHICLE
0101              1 A   DEVELOPMENT
010103            1 A     PROTOTYPE PRODUCTION
01010301          1 A       PROTOTYPE CONTRACTS
0101030101        1 A        DIRECT LABOR
010103010101      1 A          RDT-E COSTS
010103010102      1 A           PEMA COSTS
0102              1 A   INVESTMENT NON-RECURRING
010202            1 A    TOOLING
0103              1 A   INVESTMENT RECURRING
010303            1 A      QUALITY CONTROL
01030302          1 A      IN-HOUSE
0103030201        1 A         DIRECT LABOR
0103030202        1 A         MATERIALS
010305            1 A       PROCUREMENT COSTS OF VEHICLES
0104              1 A OPERATING  COSTS
010401            1 A   PERSONNEL
01040101          1 A       IN HOUSE
0104010101        1 A         DIRECT LABOR
010401010102      1 A          PEMA
```

Figure 13 A Cards From Sample



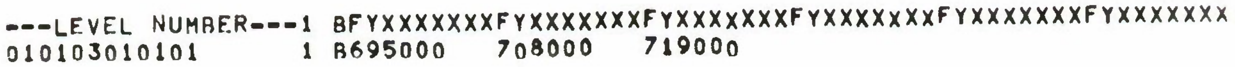
```
---LEVEL NUMBER---1 BALL XXXXXXX
010202             1 BALL 1500
0103030201         1 BALL 9000
0103030202         1 BALL  500
0104010101         1 BALL   10
```

Figure 14   B-All Cards From Sample



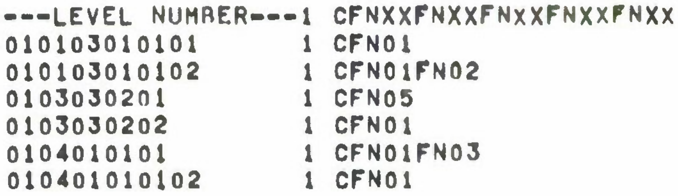
```
---LEVEL NUMBER---1 BFROM FY TO FY XXXXXXX
010401010102      1 BFROM 69 TO 80   900
010401010102      2 BFROM 81 TO 89   800
```

Figure 15   B-From Cards From Sample



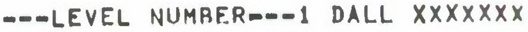
```
---LEVEL NUMBER---1 BFYXXXXXXXFYXXXXXXXFYXXXXXXXFYXXXXXXXFYXXXXXXXFYXXXXXXX
010103010101      1 B695000    708000    719000
```

Figure 16   B-Year Cards From Sample



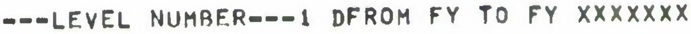
```
---LEVEL NUMBER---1 CFNXXFNXXFNXXFNXXFNXX
010103010101       1 CFN01
010103010102       1 CFN01FN02
0103030201         1 CFN05
0103030202         1 CFN01
0104010101         1 CFN01FN03
010401010102       1 CFN01
```

Figure 17 C Cards From Sample



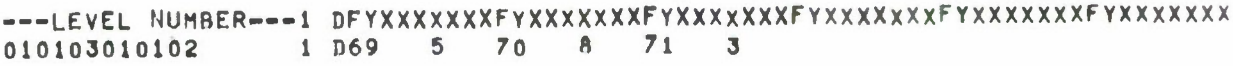
```
---LEVEL NUMBER---1 DALL XXXXXXX
```

Figure 18 D-All Cards From Sample



```
---LEVEL NUMBER---1 DFROM FY TO FY XXXXXXX
```

Figure 19 D-From Cards From Sample



```
---LEVEL NUMBER---1 DFYXXXXXXXFYXXXXXXXFYXXXXXXXFYXXXXXXXFYXXXXXXXFYXXXXXXX
010103010102       1 D69    5    70    8    71    3
```

Figure 20 D-Year Cards From Sample

```
---LEVEL NUMBER---1 EFNXXFNXXFNXXFNXXFNXX
010202            1 EFN01FN02
```

Figure 21 E Cards From Sample

```
---LEVEL NUMBER---1 G1UNITCOSTBVALUEXXQTYSTRFNXX
010305            1 G 9000.      .15        FN04
```

Figure 22 G Cards From Sample

```
---LEVEL NUMBER---1 H1UN-1XXXXXXUN-2XXXXXXUN-3XXXXXXUN-4XXXXXXUN-5XXXXXX
010305            1 H   1   2.0  25   1.0
```

Figure 23 H Cards From Sample

```
SAMPLE  CASE-TIME PHASED LIFE CYCF COST ESTIMATING MOOEL
                                          COST DATA IN MILLIONS
FY69      FY72      FY75      FY7A      FY81      FY84      FY87                                       TOTAL
 FY70      FY73      FY76      FY79      FY82      FY85      FY88
  FY71      FY74      FY77      FY80      FY83      FY86

SAMPLE CASE LIFE CYCLE COST ESTIMATING MOOEL
          X-4210 VEHICLE                                                                           01
   26.494     11.536     11.732     11.694     14.574     14.560     14.550
    35.559     11.567     11.716     14.686     14.569     14.556     14.547
     26.507     11.756     11.704     14.680     14.564     14.553                                      316.104

 DEVELOPMENT                                                                                       0101
   15.000     0.000      0.000      0.000      0.000      0.000      0.000
    24.000     0.000      0.000      0.000      0.000      0.000      0.000
     15.000     0.000      0.000      0.000      0.000      0.000                                      54.000

   PROTOTYPE PRODUCTION                                                                            010103
   15.000     0.000      0.000      0.000      0.000      0.000      0.000
    24.000     0.000      0.000      0.000      0.000      0.000      0.000
     15.000     0.000      0.000      0.000      0.000      0.000                                      54.000

     PROTOTYPF CONTRACTS                                                                           01010301
   15.000     0.000      0.000      0.000      0.000      0.000      0.000
    24.000     0.000      0.000      0.000      0.000      0.000      0.000
     15.000     0.000      0.000      0.000      0.000      0.000                                      54.000

      OIRECT LABOR                                                                                 0101030101
   15.000     0.000      0.000      0.000      0.000      0.000      0.000
    24.000     0.000      0.000      0.000      0.000      0.000      0.000
     15.000     0.000      0.000      0.000      0.000      0.000                                      54.000

        ROT-E COSTS                                                                                010103010101
    5.000     0.000      0.000      0.000      0.000      0.000      0.000
    8.000     0.000      0.000      0.000      0.000      0.000      0.000
     9.000     0.000      0.000      0.000      0.000      0.000                                      22.000

         PEMA COSTS                                                                                010103010102
   10.000     0.000      0.000      0.000      0.000      0.000      0.000
   16.000     0.000      0.000      0.000      0.000      0.000      0.000
     6.000     0.000      0.000      0.000      0.000      0.000                                      32.000

 INVESTMENT NON-RECURRING                                                                          0102
    4.500     4.500      4.500      4.500      4.500      4.500      4.500
    4.500     4.500      4.500      4.500      4.500      4.500      4.500
     4.500     4.500      4.500      4.500      4.500      4.500                                      90.000

  TOOLING                                                                                          010202
    4.500     4.500      4.500      4.500      4.500      4.500      4.500
    4.500     4.500      4.500      4.500      4.500      4.500      4.500
     4.500     4.500      4.500      4.500      4.500      4.500                                      90.000

 INVESTMENT RFCURRING                                                                              0103
    1.094     1.136      1.332      1.294      1.274      1.260      1.250
    1.159     1.167      1.316      1.286      1.269      1.256      1.247
     1.107     1.356      1.304      1.280      1.264      1.253                                      24.904
```

Figure 24   Standard Cost by Level

```
SAMPLE  CASE-TIME PHASED LIFE CYCE COST ESTIMATING MODEL
                                                COST DATA IN MILLIONS
FY69        FY72        FY75        FY78        FY81        FY84        FY87                                                    TOTAL
 FY70        FY73        FY76        FY79        FY82        FY85        FY88
  FY71        FY74        FY77        FY80        FY83        FY86

    QUALITY CONTROL                                                                                                      010303
    0.950        0.950        0.950        0.950        0.950        0.950        0.950
     0.950        0.950        0.950        0.950        0.950        0.950        0.950
      0.950        0.950        0.950        0.950        0.950        0.950                                                19.000

    IN-HOUSE                                                                                                             01030302
    0.950        0.950        0.950        0.950        0.950        0.950        0.950
     0.950        0.950        0.950        0.950        0.950        0.950        0.950
      0.950        0.950        0.950        0.950        0.950        0.950                                                19.000

       DIRECT LABOR                                                                                                      0103030201
    0.450        0.450        0.450        0.450        0.450        0.450        0.450
     0.450        0.450        0.450        0.450        0.450        0.450        0.450
      0.450        0.450        0.450        0.450        0.450        0.450                                                 9.000

       MATERIALS                                                                                                         0103030202
    0.500        0.500        0.500        0.500        0.500        0.500        0.500
     0.500        0.500        0.500        0.500        0.500        0.500        0.500
      0.500        0.500        0.500        0.500        0.500        0.500                                                10.000

     PROCUREMENT COSTS OF VEHICLES                                                                                       010305
    0.144        0.186        0.382        0.344        0.324        0.310        0.300
     0.209        0.217        0.366        0.336        0.319        0.306        0.297
      0.157        0.406        0.354        0.330        0.314        0.303                                                 5.904

OPERATING  COSTS                                                                                                         0104
    5.900        5.900        5.900        5.900        8.800        8.800        8.800
     5.900        5.900        5.900        8.900        8.800        8.800        8.800
      5.900        5.900        5.900        8.900        8.800        8.800                                               147.200

  PERSONNEL                                                                                                              010401
    5.900        5.900        5.900        5.900        8.800        8.800        8.800
     5.900        5.900        5.900        8.900        8.800        8.800        8.800
      5.900        5.900        5.900        8.900        8.800        8.800                                               147.200

     IN HOUSE                                                                                                            01040101
    5.900        5.900        5.900        5.900        8.800        8.800        8.800
     5.900        5.900        5.900        8.900        8.800        8.800        8.800
      5.900        5.900        5.900        8.900        8.800        8.800                                               147.200

       DIRECT LABOR                                                                                                      0104010101
    5.900        5.900        5.900        5.900        8.800        8.800        8.800
     5.900        5.900        5.900        8.900        8.800        8.800        8.800
      5.900        5.900        5.900        8.900        8.800        8.800                                               147.200

        PEMA                                                                                                             010401010102
    0.900        0.900        0.900        0.900        0.800        0.800        0.800
     0.900        0.900        0.900        0.900        0.800        0.800        0.800
      0.900        0.900        0.900        0.900        0.800        0.800                                                17.200
```

Figure 25   Standard Cost by Level

```
SAMPLE  CASE-TIME PHASED LIFE CYCE COST ESTIMATING MODEL
FY69         FY72         FY75         FY78         FY81         FY84         FY87                                                   TOTAL
 FY70         FY73         FY76         FY79         FY82         FY85         FY88
  FY71         FY74         FY77         FY80         FY83         FY86

SUMMARY BY MAJOR COST CATEGORY
   26.494       11.536       11.732       11.694       14.574       14.560       14.550
    35.559       11.567       11.716       14.686       14.569       14.556       14.547
     26.507       11.756       11.704       14.680       14.564       14.553                                                      316.104

DEVELOPMENT
   15.000        0.000        0.000        0.000        0.000        0.000        0.000
    24.000        0.000        0.000        0.000        0.000        0.000        0.000
     15.000        0.000        0.000        0.000        0.000        0.000                                                       54.000

INVESTMENT-NON RECURRING
    4.500        4.500        4.500        4.500        4.500        4.500        4.500
     4.500        4.500        4.500        4.500        4.500        4.500        4.500
      4.500        4.500        4.500        4.500        4.500        4.500                                                       90.000

INVESTMENT RECURRING
    1.094        1.136        1.332        1.294        1.274        1.260        1.250
     1.159        1.167        1.316        1.286        1.269        1.256        1.247
      1.107        1.356        1.304        1.280        1.264        1.253                                                       24.904

OPERATING
    5.900        5.900        5.900        5.900        8.800        8.800        8.800
     5.900        5.900        5.900        8.900        8.800        8.800        8.800
      5.900        5.900        5.900        8.900        8.800        8.800                                                      147.200
```

Figure 26 Summary by Major Cost Category

```
SAMPLE  CASE-TIME PHASED LIFE CYCE COST ESTIMATING MODEL
FY69        FY72        FY75        FY78        FY81        FY84        FY87                                              TOTAL
 FY70        FY73        FY76        FY79        FY82        FY85        FY88
  FY71        FY74        FY77        FY80        FY83        FY86

SUMMARY BY APPROPRIATION
    15.900      0.900       0.900       0.900       0.800       0.800       0.800
     24.900      0.900       0.900       0.900       0.800       0.800       0.800
      15.900      0.900       0.900       0.900       0.800       0.800                                                   71.200

  RDT-E
    5.000       0.000       0.000       0.000       0.000       0.000       0.000
     8.000       0.000       0.000       0.000       0.000       0.000       0.000
      9.000       0.000       0.000       0.000       0.000       0.000                                                   22.000

  PEMA
    10.900      0.900       0.900       0.900       0.800       0.800       0.800
     16.900      0.900       0.900       0.900       0.800       0.800       0.800
      6.900       0.900       0.900       0.900       0.800       0.800                                                   49.200
```

Figure 27 Summary by Appropriation

```
SAMPLE  CASE-TIME PHASED LIFE CYCF COST ESTIMATING MODEL
FY69       FY72       FY75       FY78       FY81       FY84       FY87                                                      TOTAL
 FY70       FY73       FY76       FY79       FY82       FY85       FY88
  FY71       FY74       FY77       FY80       FY83       FY86

SUMMARY-COST CATEGORY BY APPROPRIATION
     15.900     0.900      0.900      0.900      0.800      0.800      0.800
      24.900     0.900      0.900      0.900      0.800      0.800      0.800
       15.900     0.900      0.900      0.900      0.800      0.800                                                          71.200

DEVELOPMENT

RDT-E
    5.000      0.000      0.000      0.000      0.000      0.000      0.000
     8.000      0.000      0.000      0.000      0.000      0.000      0.000
      9.000      0.000      0.000      0.000      0.000      0.000                                                          22.000

PEMA
   10.000      0.000      0.000      0.000      0.000      0.000      0.000
    16.000      0.000      0.000      0.000      0.000      0.000      0.000
      6.000      0.000      0.000      0.000      0.000      0.000                                                          32.000

OPERATING

PEMA
    0.900      0.900      0.900      0.900      0.800      0.800      0.800
     0.900      0.900      0.900      0.900      0.800      0.800      0.800
      0.900      0.900      0.900      0.900      0.800      0.800                                                          17.200
```

Figure 28   Summary of Cost Category by Appropriation

Next page is blank.

APPENDIX

FLOW CHARTS AND PROGRAM LISTINGS

This appendix contains flow charts and listings of each program in the sequence (see Figures A.1 and A.2).

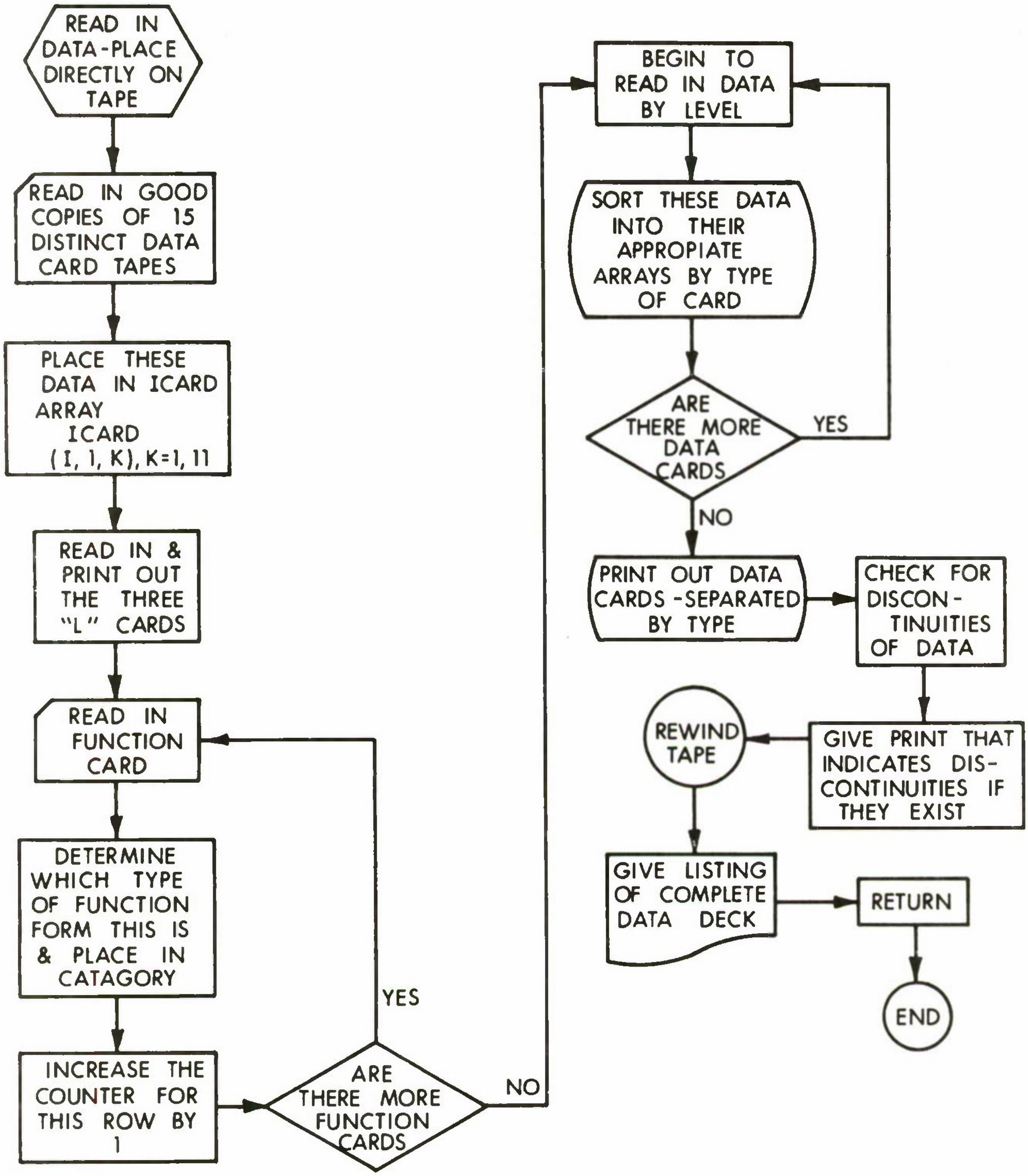

READ IN DATA-PLACE DIRECTLY ON TAPE
READ IN GOOD COPIES OF 15 DISTINCT DATA CARD TAPES
PLACE THESE DATA IN ICARD ARRAY ICARD (I, 1, K), K=1, 11
READ IN & PRINT OUT THE THREE "L" CARDS
READ IN FUNCTION CARD
DETERMINE WHICH TYPE OF FUNCTION FORM THIS IS & PLACE IN CATAGORY
INCREASE THE COUNTER FOR THIS ROW BY 1
ARE THERE MORE FUNCTION CARDS
YES
NO
BEGIN TO READ IN DATA BY LEVEL
SORT THESE DATA INTO THEIR APPROPIATE ARRAYS BY TYPE OF CARD
ARE THERE MORE DATA CARDS
YES
NO
PRINT OUT DATA CARDS-SEPARATED BY TYPE
CHECK FOR DISCON-TINUITIES OF DATA
GIVE PRINT THAT INDICATES DIS-CONTINUITIES IF THEY EXIST
REWIND TAPE
GIVE LISTING OF COMPLETE DATA DECK
RETURN
END

Figure A-1 Flow Chart for the Data Check and Imput Tape Preparation Program

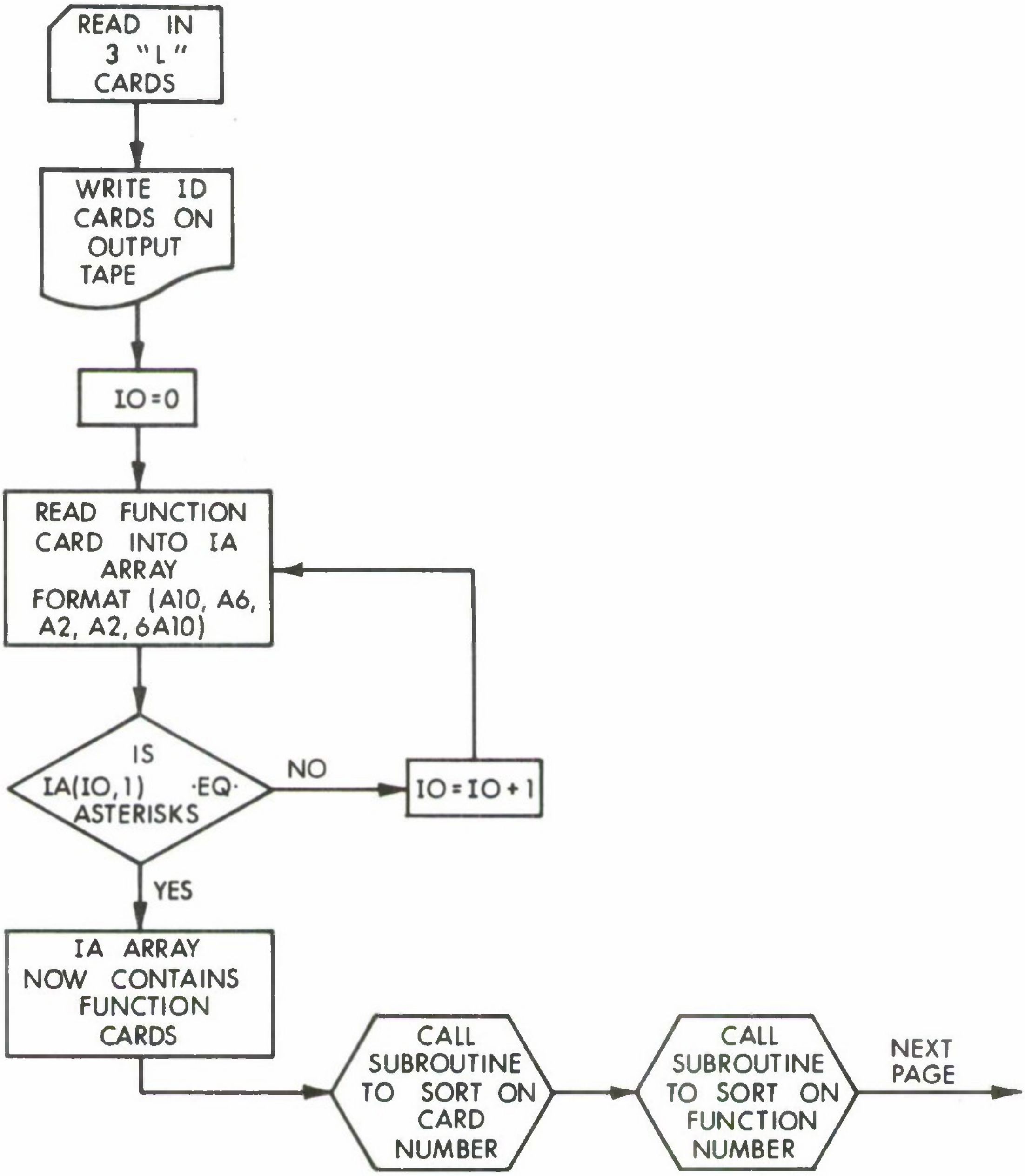

READ IN 3 "L" CARDS
WRITE ID CARDS ON OUTPUT TAPE
IO=0
READ FUNCTION CARD INTO IA ARRAY FORMAT (A10, A6, A2, A2, 6A10)
IS IA(IO,1) ·EQ· ASTERISKS
NO
IO=IO+1
YES
IA ARRAY NOW CONTAINS FUNCTION CARDS
CALL SUBROUTINE TO SORT ON CARD NUMBER
CALL SUBROUTINE TO SORT ON FUNCTION NUMBER
NEXT PAGE

Figure A-2 Chart I – Flow Charts for Optional Program Which Sorts Data into Correct Order.

THIS PROGRAM ASSUMES THE FOLLOWING ORDER OF THE IMPUT CARDS:

1. THE 3 L CARDS (IN ORDER)
2. THE FUNCTION CARDS (NOT NECESSARILY IN ORDER)
3. A CARD WITH 10 ASTERISKS
4. THE LEVEL DATA (NOT NECESSARILY IN ORDER)
5. A BLANK CARD

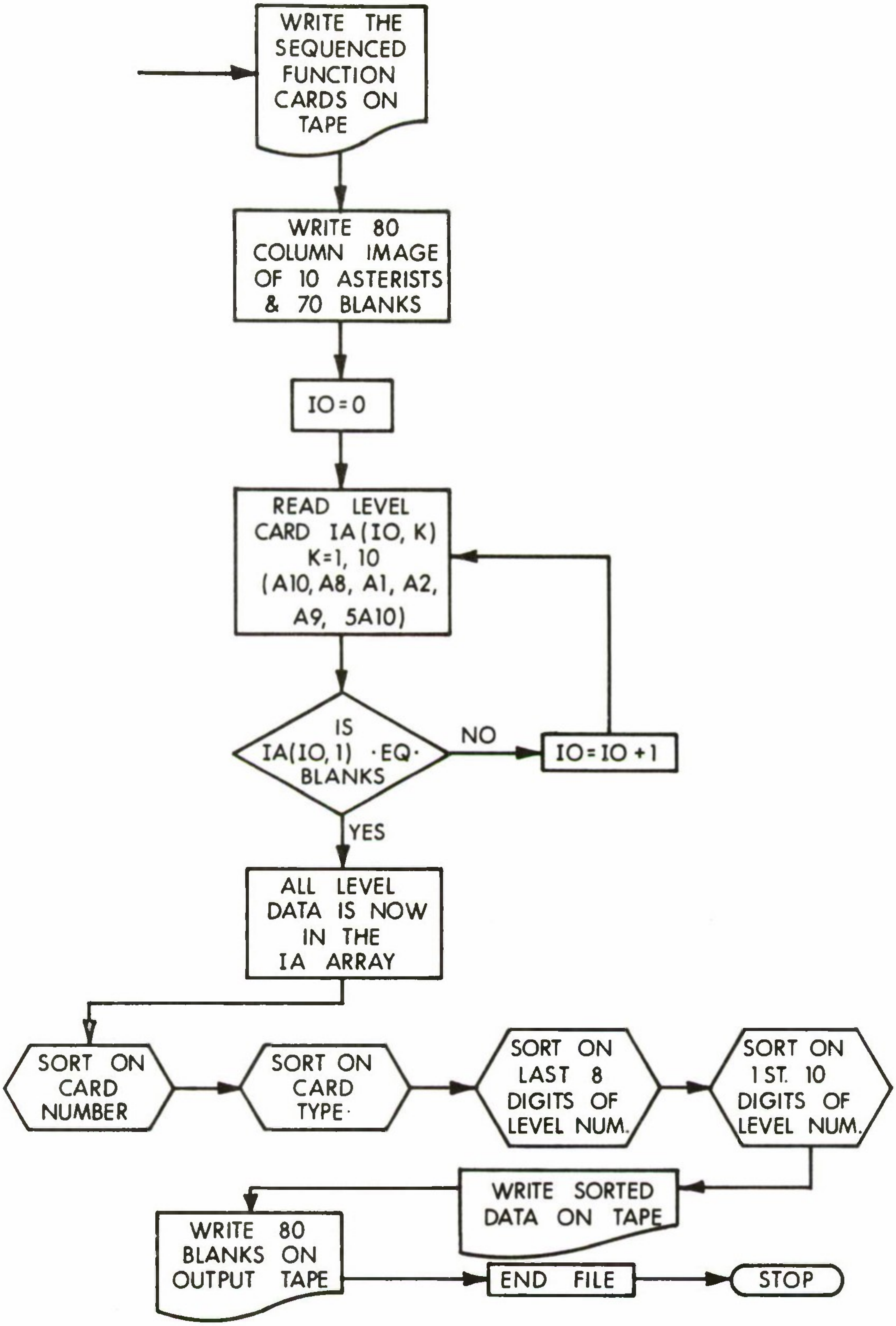

WRITE THE SEQUENCED FUNCTION CARDS ON TAPE
WRITE 80 COLUMN IMAGE OF 10 ASTERISTS & 70 BLANKS
IO=0
READ LEVEL CARD IA(IO, K) K=1, 10 (A10, A8, A1, A2, A9, 5A10)
IS IA(IO, 1) ·EQ· BLANKS
NO
IO=IO+1
YES
ALL LEVEL DATA IS NOW IN THE IA ARRAY
SORT ON CARD NUMBER
SORT ON CARD TYPE
SORT ON LAST 8 DIGITS OF LEVEL NUM.
SORT ON 1 ST. 10 DIGITS OF LEVEL NUM.
WRITE SORTED DATA ON TAPE
WRITE 80 BLANKS ON OUTPUT TAPE
END FILE
STOP

Figure A-2 Chart II

A listing of the Data Check and Input Tape Preparation Program (Program 1 in the Sequence) Follows. (See Figures A.3 through A.7).

```
 FEB. 7,72  BRLESC2 FORTRAN.                                    CB 17..29
*      SA11WC 392 107 4210 COPES  PROGRAM 1

       DIMENSION ICARD(14,100,8),ITEMP(8),NCOUNT(14),IZ(18),IZ1(18)
       DIMENSION ITEMP1(8),PLANK(8)
     1 FORMAT(A10,A8,A1,A2,A1,A8,5A10)
     2 FORMAT(8A10)
     3 FORMAT(70X,A10)
     4 FORMAT(2A10,2A1,5A10,A8)
     5 FORMAT(18A1)
     6 FORMAT (1X,A1)
     7 FORMAT(1H1)
     8 FORMAT (2A1)
     9 FORMAT(1H-,'A DISCONTINUITY EXISTS BETWEEN LEVELS',3X,18A1,3X,'AND
      1',3X,18A1)
       DATA(ICARD(1,1,K),K=1,4) /'          ','    FNXX1 ','FALL XXXXX',
      1'XX        '/
       DATA(ICARD(2,1,K),K=1,5) /'          ','    FNXX1 ','FFROM FY T',
      1'O FY XXXXX','XX        '/
       DATA(ICARD(3,1,K),K=2,8) /'    FNXX1 ','FFYXXXXXXX','FYXXXXXXXX',
      1'YXXXXXXXFY','XXXXXXXFYX','XXXXXXFYXX','XXXXX     '/
       DATA(ICARD(4,1,K),K=1,8) /'---LEVEL N','UMBER---1 ','A---------',
      1'----LEVEL ','IDENTIFICA','TION NAME-','----------','----------'/
       DATA(ICARD(5,1,K),K=1,4) /'---LEVEL N','UMBER---1 ','BALL XXXXX',
      1'XX        '/
       DATA(ICARD(6,1,K),K=1,5) /'---LEVEL N','UMBER---1 ','BFROM FY T',
      1'O FY XXXXX','XX        '/
       DATA(ICARD(7,1,K),K=1,8) /'---LEVEL N','UMBER---1 ','BFYXXXXXXX',
      1'FYXXXXXXXF','YXXXXXXXFY','XXXXXXXFYX','XXXXXXFYXX','XXXXX     '/
       DATA(ICARD(8,1,K),K=1,5) /'---LEVEL N','UMBER---1 ','CFNXXFNXXF',
      1'NXXFNXXFNX','X         '/
       DATA(ICARD(9,1,K),K=1,4) /'---LEVEL N','UMBER---1 ','DALL XXXXX',
      1'XX        '/
       DATA(ICARD(10,1,K),K=1,5)/'---LEVEL N','UMBER---1 ','DFROM FY T',
      1'O FY XXXXX','XX        '/
       DATA(ICARD(11,1,K),K=1,8)/'---LEVEL N','UMBER---1 ','DFYXXXXXXX',
      1'FYXXXXXXXF','YXXXXXXXFY','XXXXXXXFYX','XXXXXXFYXX','XXXXX     '/
       DATA(ICARD(12,1,K),K=1,5)/'---LEVEL N','UMBER---1 ','EFNXXFNXXF',
      1'NXXFNXXFNX','X         '/
       DATA(ICARD(13,1,K),K=1,5)/'---LEVEL N','UMBER---1 ','G1UNITCOST',
      1'BVALUEXXQT','YSTRFNXX  '/
       DATA(ICARD(14,1,K),K=1,8)/'---LEVEL N','UMBER---1 ','H1UN-1XXXX',
      1'XXUN-2XXXX','XXUN-3XXXX','XXUN-4XXXX','XXUN-5XXXX','XX        '/
       DATA BLANK/10H          /
       DATA LANK/10H         /
       DATA ITAR/10H***********/
       DATA STAR/10H***********/
       REWIND 1
       DO 12 I=1,8
       PLANK(I)=BLANK
    12 CONTINUE
       DO 10 I=1,14
       NCOUNT(I)=1
    10 CONTINUE
       DO 15 I=1,3
       READ 2, (ITEMP(J),J=1,8)
       WRITE (1,2) (ITEMP(J),J=1,8)
    15 CONTINUE
       DO 17 I=1,300
       READ 2, (ITEMP(J),J=1,8)
      IF (ITEMP(1).EQ.ITAR) GOTO 18
      WRITE (1,2) (ITEMP(J),J=1,8)
   17 CONTINUE
   18 WRITE (1,2) STAR, (PLANK(J),J=1,7)
      ISTART=3+I
      DO 20 KI=ISTART,50000
      READ 2,(ITEMP(I),I=1,8)
      IF (ITEMP(1).EQ. LANK) GOTO 25
      WRITE (1,2) (ITEMP(I),I=1,8)
   20 CONTINUE
C*****
C             ALL DATA NOW WRITTEN ON UNIT 1
C*****
   25 WRITE(1,2) (PLANK(I),I=1,8)
      END FILE 1
      REWIND 1
      DO 30 KL=1,3
      READ(1,2)(ITEMP(I),I=1,8)
      PRINT 2, (ITEMP(I),I=1,8)
   30 CONTINUE
      DO 100 KL=1,300
      READ(1,2)(ITEMP(I),I=1,8)
      IF (ITEMP(1).EQ.ITAR) GOTO 110
      DECODE(80,6,ITEMP(3)) CHK
      IF (CHK.EQ.1HA) GOTO 80
      IF (CHK.EQ.1HF) GOTO 50
      NCOUNT(3)=NCOUNT(3)+1
      N3=NCOUNT(3)
      DO 40 I=1,8
      ICARD(3,N3,I)=ITEMP(I)
   40 CONTINUE
      GOTO 100
   50 NCOUNT(2)=NCOUNT(2)+1
      N2=NCOUNT(2)
      DO 60 I=1,8
      ICARD(2,N2,I)=ITEMP(I)
   60 CONTINUE
      GOTO 100
   80 NCOUNT(1)=NCOUNT(1)+1
      N1=NCOUNT(1)
      DO 90 I=1,8
      ICARD(1,N1,I)=ITEMP(I)
   90 CONTINUE
  100 CONTINUE
  110 CONTINUE
      READ(1,2)(ITEMP(I),I=1,8)
      DECODE (80,5,ITEMP(1)) (IZ(I),I=1,18)
      CALL DEGREE(IZ,LEV,1,18)
      DO 500 JKL=1,50000
      DECODE (80,8,ITEMP(3)) CHK1,CHK2
  130 IF(CHK1.EQ.1HA) GOTO 150
      IF(CHK1.EQ.1HB) GOTO 200
      IF(CHK1.EQ.1HC) GOTO 250
      IF(CHK1.EQ.1HD) GOTO 300
      IF(CHK1.EQ.1HE) GOTO 350
      IF(CHK1.EQ.1HG) GOTO 400
      IF(CHK1.EQ.1HH) GOTO 450
  150 NCOUNT(4)=NCOUNT(4)+1
      N4=NCOUNT(4)
      DO 160 I=1,8
      ICARD(4,N4,I)=ITEMP(I)
  160 CONTINUE
      GOTO 480
  200 IF(CHK2.EQ.1HA) GOTO 210
      IF(CHK2.EQ.1HF) GO TO 220
      NCOUNT(7)=NCOUNT(7)+1
      N7=NCOUNT(7)
      DO 205 I=1,8
      ICARD(7,N7,I)=ITEMP(I)
  205 CONTINUE
      GOTO 480
  210 NCOUNT(5)=NCOUNT(5)+1
      N5=NCOUNT(5)
      DO 215 I=1,8
      ICARD(5,N5,I)=ITEMP(I)
  215 CONTINUE
      GOTO 480
  220 NCOUNT(6)=NCOUNT(6)+1
      N6=NCOUNT(6)
      DO 225 I=1,8
      ICARD(6,N6,I)=ITEMP(I)
  225 CONTINUE
      GOTO 480
  250 NCOUNT(8)=NCOUNT(8)+1
      N8=NCOUNT(8)
      DO 295 I=1,8
      ICARD(8,N8,I)=ITEMP(I)
  295 CONTINUE
      GOTO 480
  300 IF(CHK2.EQ.1HA) GOTO 310
      IF(CHK2.EQ.1HF) GOTO 320
      NCOUNT(11)=NCOUNT(11)+1
      N11=NCOUNT(11)
      DO 305 I=1,8
      ICARD(11,N11,I)=ITEMP(I)
  305 CONTINUE
      GOTO 480
  310 NCOUNT(9)=NCOUNT(9)+1
      N9=NCOUNT(9)
      DO 315 I=1,8
      ICARD(9,N9,I)=ITEMP(I)
  315 CONTINUE
      GOTO 480
  320 NCOUNT(10)=NCOUNT(10)+1
      N10=NCOUNT(10)
      DO 345 I=1,8
      ICARD(10,N10,I)=ITEMP(I)
  345 CONTINUE
      GOTO 480
  350 NCOUNT(12)=NCOUNT(12)+1
      N12=NCOUNT(12)
      DO 395 I=1,8
      ICARD(12,N12,I)=ITEMP(I)
  395 CONTINUE
      GOTO 480
  400 NCOUNT(13)=NCOUNT(13)+1
      N13=NCOUNT(13)
      DO 405 I=1,8
      ICARD(13,N13,I)=ITEMP(I)
  405 CONTINUE
```

```
      GOTO 480
  450 NCOUNT(14)=NCOUNT(14)+1
      N14=NCOUNT(14)
      DO 475 I=1,8
      ICARD(14,N14,I)=ITEMP(I)
  475 CONTINUE
  480 READ (1,2) (ITEMP1(I),I=1,8)
      IF (ITEMP1(1).EQ.LANK) GOTO 510
      DECODE (80,5,ITEMP1(1))(IZ1(I),I=1,18)
      DO 485 K=1,18
      IF(IZ(K).NE.IZ1(K))GOTO 488
  485 CONTINUE
  486 DO 487 K=1,8
      ITEMP(K)=ITEMP1(K)
  487 CONTINUE
      GOTO 500
  488 CALL DEGREE(IZ1,LEV1,1,18)
      NDIF=LEV1-LEV
      IF(NDIF.GE.2)GOTO 490
  491 DO 489 K=1,8
      ITEMP(K)=ITEMP1(K)
  489 CONTINUE
      LEV=LEV1
      GO TO 500
  490 PRINT 9,((IZ(I),I=1,18),(IZ1(I),I=1,18))
      GO TO 491
  500 CONTINUE
  510 DO 600 JK=1,14
      PRINT 7
      IKNOW=NCOUNT(JK)
      DO 550 JOB=1,IKNOW
      PRINT 2, (ICARD(JK,JOB,M),M=1,8)
  550 CONTINUE
  600 CONTINUE
      REWIND 1
      STOP
      END
      SUBROUTINE DEGREE (IARRAY,NDEG,ISTART,ISTOP)
      DIMENSION IARRAY(18)
      DO 10 I=ISTART,ISTOP
      IF (IARRAY(I).EQ.1H ) GO TO 15
   10 CONTINUE
      NDEG=9
      GO TO 20
   15 NDEG=(I-1)/2
   20 CONTINUE
      RETURN
      END
*     LIST(STOP)
```

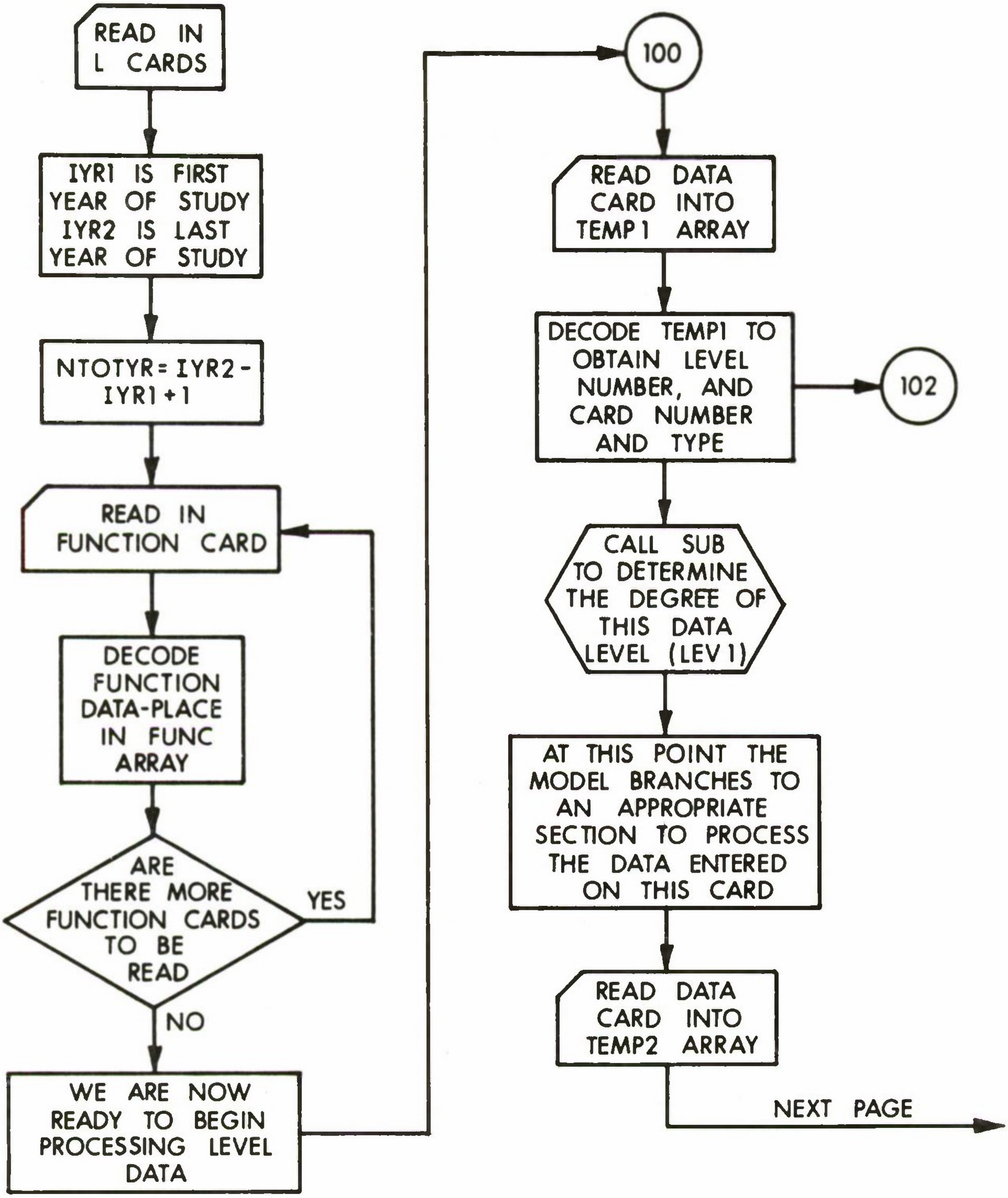

READ IN L CARDS
IYR1 IS FIRST YEAR OF STUDY IYR2 IS LAST YEAR OF STUDY
NTOTYR = IYR2 - IYR1 + 1
READ IN FUNCTION CARD
DECODE FUNCTION DATA-PLACE IN FUNC ARRAY
ARE THERE MORE FUNCTION CARDS TO BE READ
YES
NO
WE ARE NOW READY TO BEGIN PROCESSING LEVEL DATA
100
READ DATA CARD INTO TEMP1 ARRAY
DECODE TEMP1 TO OBTAIN LEVEL NUMBER, AND CARD NUMBER AND TYPE
102
CALL SUB TO DETERMINE THE DEGREE OF THIS DATA LEVEL (LEV1)
AT THIS POINT THE MODEL BRANCHES TO AN APPROPRIATE SECTION TO PROCESS THE DATA ENTERED ON THIS CARD
READ DATA CARD INTO TEMP2 ARRAY
NEXT PAGE

Figure A-3 Chart I

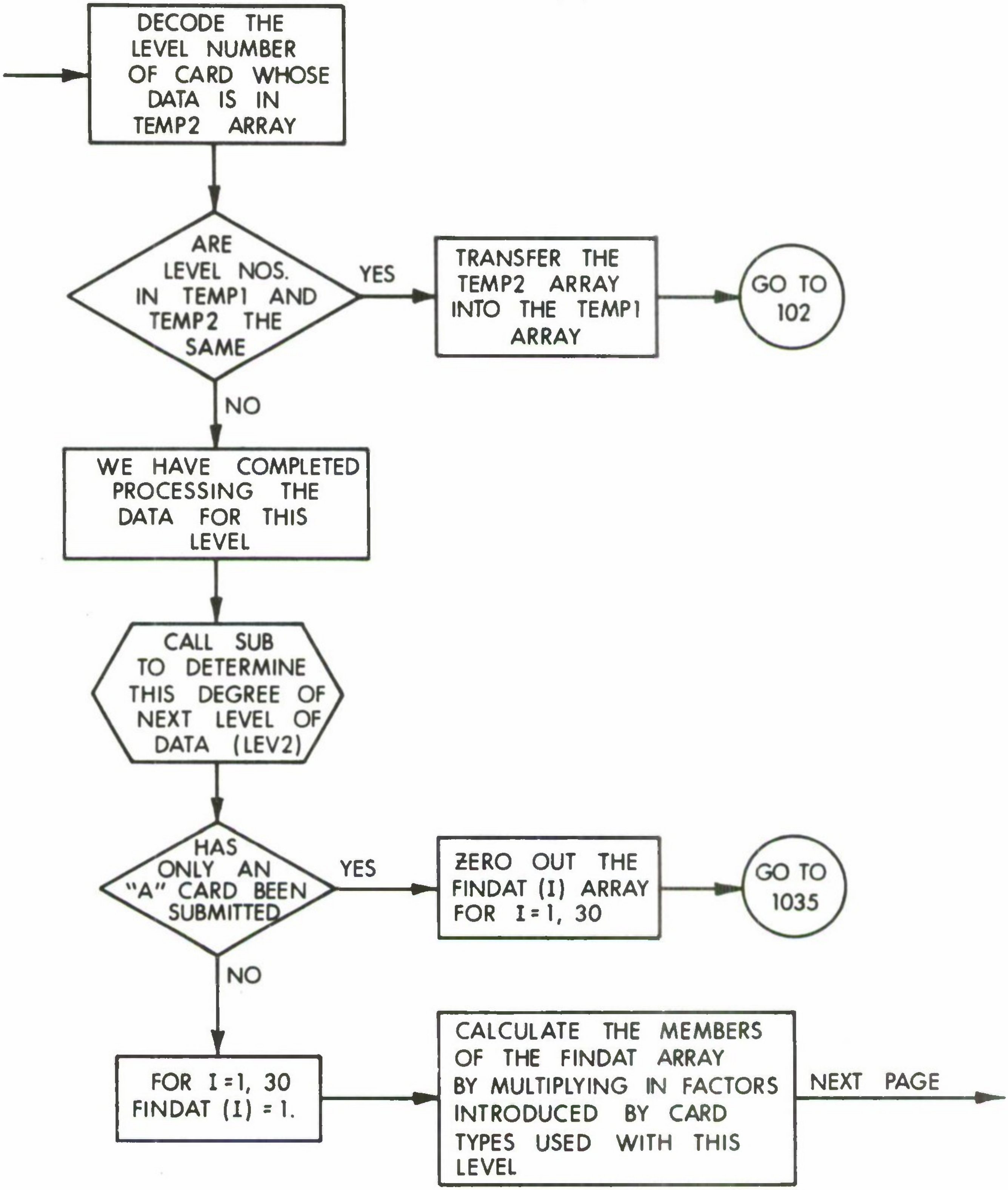

DECODE THE LEVEL NUMBER OF CARD WHOSE DATA IS IN TEMP2 ARRAY
ARE LEVEL NOS. IN TEMP1 AND TEMP2 THE SAME
YES
TRANSFER THE TEMP2 ARRAY INTO THE TEMP1 ARRAY
GO TO 102
NO
WE HAVE COMPLETED PROCESSING THE DATA FOR THIS LEVEL
CALL SUB TO DETERMINE THIS DEGREE OF NEXT LEVEL OF DATA (LEV2)
HAS ONLY AN "A" CARD BEEN SUBMITTED
YES
ZERO OUT THE FINDAT (I) ARRAY FOR I = 1, 30
GO TO 1035
NO
FOR I = 1, 30 FINDAT (I) = 1.
CALCULATE THE MEMBERS OF THE FINDAT ARRAY BY MULTIPLYING IN FACTORS INTRODUCED BY CARD TYPES USED WITH THIS LEVEL
NEXT PAGE

Figure A-3 Chart II

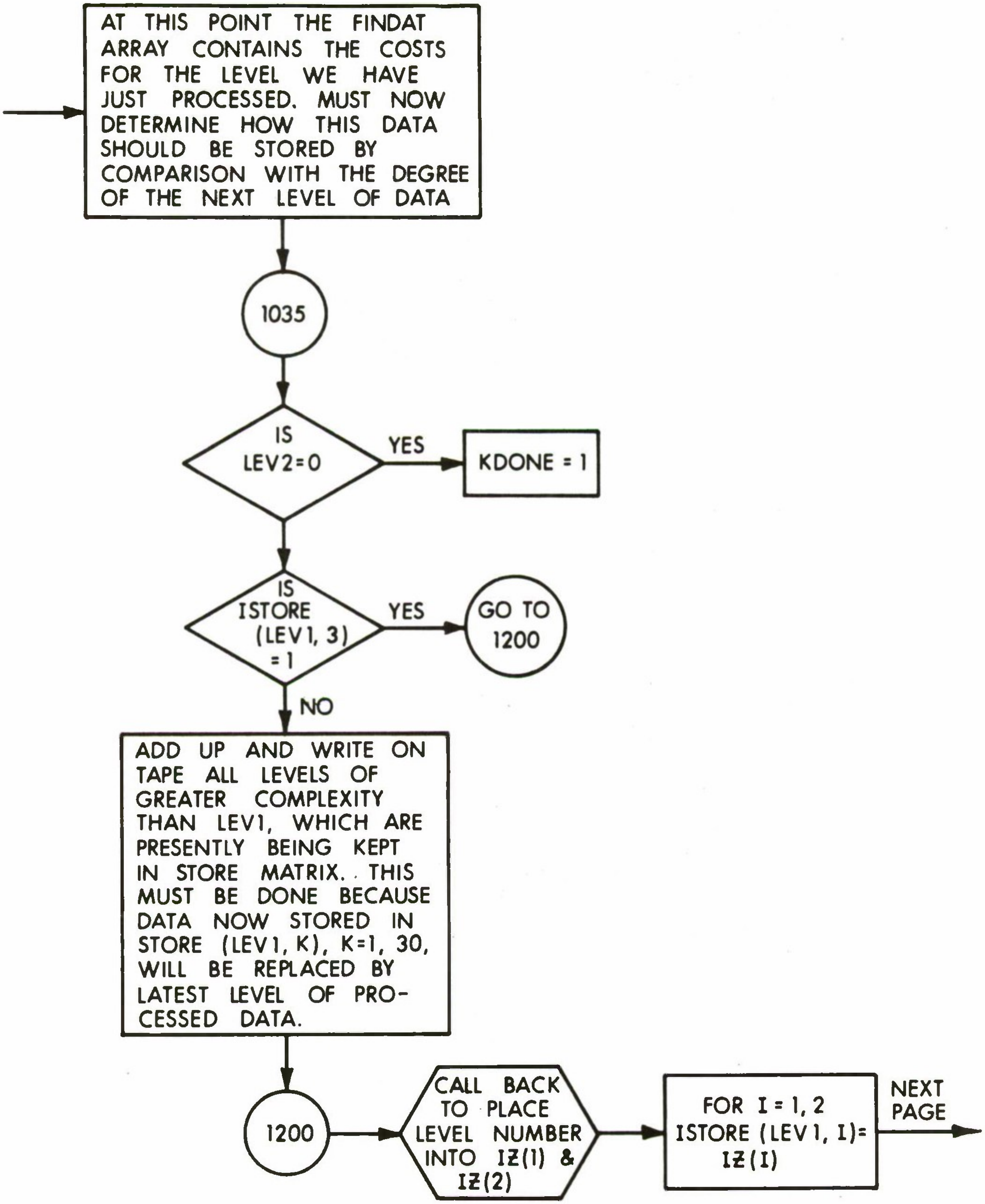

AT THIS POINT THE FINDAT ARRAY CONTAINS THE COSTS FOR THE LEVEL WE HAVE JUST PROCESSED. MUST NOW DETERMINE HOW THIS DATA SHOULD BE STORED BY COMPARISON WITH THE DEGREE OF THE NEXT LEVEL OF DATA
1035
IS LEV2=0
YES
KDONE = 1
IS ISTORE (LEV1, 3) = 1
YES
GO TO 1200
NO
ADD UP AND WRITE ON TAPE ALL LEVELS OF GREATER COMPLEXITY THAN LEV1, WHICH ARE PRESENTLY BEING KEPT IN STORE MATRIX. THIS MUST BE DONE BECAUSE DATA NOW STORED IN STORE (LEV1, K), K=1, 30, WILL BE REPLACED BY LATEST LEVEL OF PROCESSED DATA.
1200
CALL BACK TO PLACE LEVEL NUMBER INTO IZ(1) & IZ(2)
FOR I = 1, 2 ISTORE (LEV1, I)= IZ(I)
NEXT PAGE

Figure A-3 Chart III

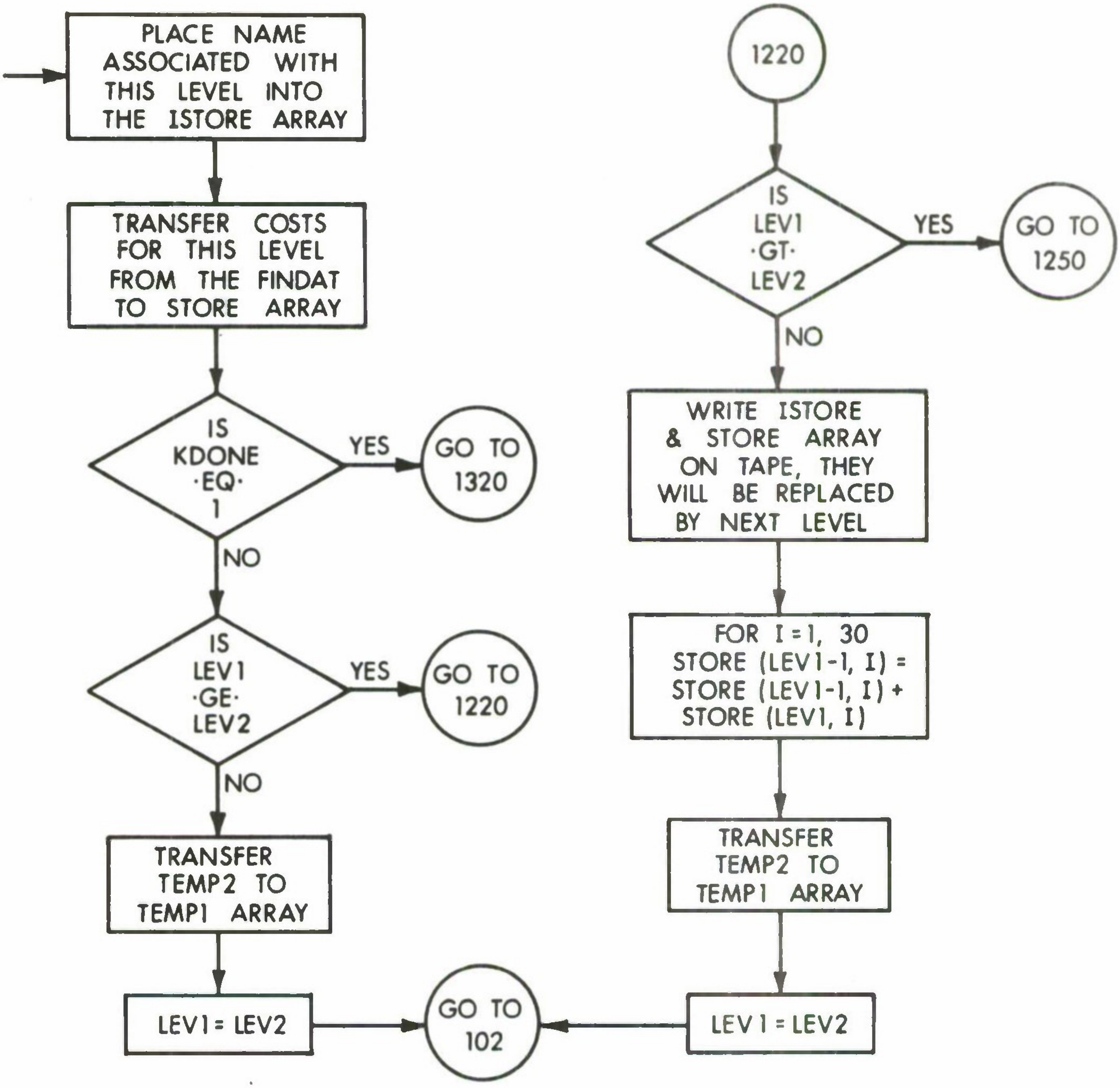

PLACE NAME ASSOCIATED WITH THIS LEVEL INTO THE ISTORE ARRAY
TRANSFER COSTS FOR THIS LEVEL FROM THE FINDAT TO STORE ARRAY
IS KDONE ·EQ· 1
YES
GO TO 1320
NO
IS LEV1 ·GE· LEV2
YES
GO TO 1220
NO
TRANSFER TEMP2 TO TEMP1 ARRAY
LEV1 = LEV2
GO TO 102
1220
IS LEV1 ·GT· LEV2
YES
GO TO 1250
NO
WRITE ISTORE & STORE ARRAY ON TAPE, THEY WILL BE REPLACED BY NEXT LEVEL
FOR I = 1, 30 STORE (LEV1-1, I) = STORE (LEV1-1, I) + STORE (LEV1, I)
TRANSFER TEMP2 TO TEMP1 ARRAY
LEV1 = LEV2

Figure A-3 Chart IV

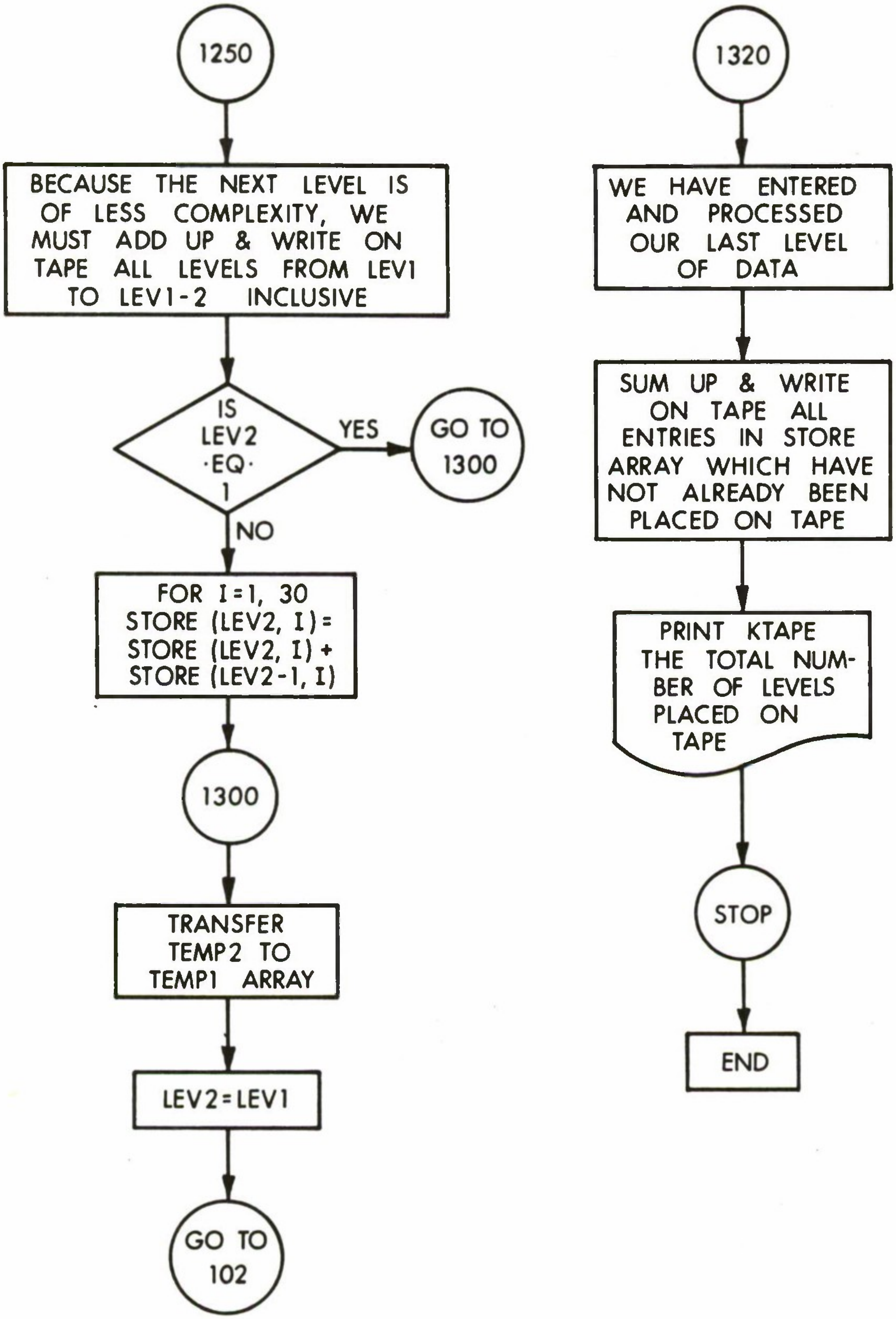

1250
BECAUSE THE NEXT LEVEL IS OF LESS COMPLEXITY, WE MUST ADD UP & WRITE ON TAPE ALL LEVELS FROM LEV1 TO LEV1-2 INCLUSIVE
IS LEV2 ·EQ· 1
YES
GO TO 1300
NO
FOR I=1, 30 STORE (LEV2, I)= STORE (LEV2, I) + STORE (LEV2-1, I)
1300
TRANSFER TEMP2 TO TEMP1 ARRAY
LEV2=LEV1
GO TO 102
1320
WE HAVE ENTERED AND PROCESSED OUR LAST LEVEL OF DATA
SUM UP & WRITE ON TAPE ALL ENTRIES IN STORE ARRAY WHICH HAVE NOT ALREADY BEEN PLACED ON TAPE
PRINT KTAPE THE TOTAL NUMBER OF LEVELS PLACED ON TAPE
STOP
END

Figure A-3 Chart V

A listing of the Main Processing Program (Program 2 in the sequence) follows. (See Figure A.8)

```
FEB. 7,72  BRLESC2 FORTRAN.                                            CB 17..31
*      SA11WC 392 107 4210  COPES PROGRAM 2

$      COMMENT USES TAPE 1 AS INPUT
$      COMMENT USES TAPE 2 AS OUTPUT
       DIMENSION REPNAM(9),FUNDAT(8),VAL(6),TEMP1(8),TEMP2(8),IFN(5),
      1    ADATA(12),BDATA(30),CDATA(30),DDATA(30),EDATA(30),GDATA(30)
       DIMENSION DATLEV(30),IZ(2),FINDAT(30),LEVEL1(18)
       DIMENSION LEVEL2(18),IYR(5)
       DIMENSION FUNC(99,30), INAME1(12), FN(5), QTY(10), NUNIT(6)
       DIMENSION ISTORE(9,15),STORE(9,30)
       DIMENSION TTEMP(24)
       DATA XER/10HXXXXX     /
       DATA BLANK/10H          /
       DATA LANK/10H          /
     1 FORMAT (8A10)
     2 FORMAT (6X,I2,3X,A1)
     3 FORMAT (1X,6(I2,F7.3))
     4 FORMAT (5X,F7.3)
     5 FORMAT (6X,I2,4X,I2,1X,F7.3)
     6 FORMAT (21X,A8,5A10)
     7 FORMAT (21X,A9,2A10,I2,1X,A9,1X,I4)
     8 FORMAT (21X,I2,2X,I2)
     9 FORMAT (18R1,A1,1X,2A1)
    10 FORMAT (1X,A9,5A10)
    11 FORMAT (3X,I2,4(2X,I2))
    12 FORMAT (1X,F9.0,F6.6,2X,I6,2X,I2)
    13 FORMAT (1X,5(I4,F6.3))
    14 FORMAT (1H1,A8,5A10,A8,2A10,/,50X,I2,2X,I2,2X,I2)
    15 FORMAT (I4,2X,I4)
    16 FORMAT (A9,5A10,A9,2A10,5X,I2,1X,A9,1X,I4)
    17 FORMAT (A10,122X)
    18 FORMAT (A10,A8,A9,5A10,A9,2A10)
    19 FORMAT (10F12.3,/,10F12.3,/,10F12.3)
    20 FORMAT (I5)
       KTAPE=0
       IB=0$IC=0$ID=0$IE=0$IG=0
       REWIND 1
       REWIND 2
       NINER=99999
       READ (1,1) (TTEMP(I),I=1,8)
       READ (1,1) (TTEMP(I),I=9,16)
       READ (1,1) (TTEMP(I),I=17,24)
       DECODE(80,6,TTEMP(1)) (REPNAM(I),I=1,6)
       DECODE(80,7,TTEMP( 9)) ((REPNAM(I),I=7,9),NDAY,NMONTH,NYR)
       DECODE(80,8,TTEMP(17)) IYR1,IYR2
       WRITE (2,16) ((REPNAM(I),I=1,9),NDAY,NMONTH,NYR)
       DO 25 I=1,9
       ISTORE(I,3)=1
    25 CONTINUE
       NTOTYR=IYR2-IYR1+1
       WRITE (2,15) IYR1,IYR2
    50 READ (1,1) (FUNDAT(I),I=1,8)
C*****          CARD WITH 10* MUST FOLLOW FUNCTION CARDS
       IF (FUNDAT(1).EQ.10H**********) GO TO 90
       DECODE (80,2,FUNDAT(2)) NUM,TYPE
       IF (TYPE.EQ.1HA) GO TO 70
       IF (TYPE.EQ.1HF) GO TO 80
       DECODE (80,3,FUNDAT(3)) ((IYR(I),VAL(I)),I=1,6)
       DO 60 I=1,6
      IF (IYR(I).EQ.10H          ) GO TO 50
      J=IYR(I)-IYR1+1
      FUNC(NUM,J)=VAL(I)
   60 CONTINUE
      GO TO 50
   70 DECODE (80,4,FUNDAT(3)) VALU
      DO 75 I=1,NTOTYR
      FUNC(NUM,I)=VALU
   75 CONTINUE
      GO TO 50
   80 DECODE (80,5,FUNDAT(3)) NSTART,NEND,VALU
      I1=NSTART-IYR1+1
      I2=NEND-IYR1+1
      DO 85 I=I1,I2
      FUNC(NUM,I)=VALU
   85 CONTINUE
      GO TO 50
C*****
C             AT THIS POINT WE ARE READY TO BEGIN PROCESSING LEVEL DATA
C*****
   90 DO 91 I=1,9
      ISTORE(I,3)=1
   91 CONTINUE
  100 READ (1,1) (TEMP1(I),I=1,8)
  102 DECODE (80,9,TEMP1(1))((LEVEL1(I),I=1,18),CARDNO,CARD,TYPE)
      CALL DEGREE (LEVEL1,LEV1,1,18)
      IF (CARD.EQ.1HA) GOTO 105
      IF (CARD.EQ.1HB) GOTO 111
      IF (CARD.EQ.1HC) GOTO 138
      IF (CARD.EQ.1HD) GOTO 160
      IF (CARD.EQ.1HE) GOTO 183
      IF (CARD.EQ.1HG) GOTO 200
C*****
C       THIS SECTION PROCESSES A TYPE DATA CARDS
C       CARNO TELLS WHETHER 1ST OR 2ND A CARD FOR THIS LEVEL
C*****
  105 IA=1
      IF (CARDNO.EQ.1H2) GO TO 110
      DECODE (80,10,TEMP1(3))(INAME1(I),I=1,6)
      GO TO 1000
  110 DECODE (80,10, TEMP1(3)) (INAME1(I),I=7,12)
      GO TO 1000
C*****
C       THIS SECTION PROCESSES B TYPE DATA CARDS
C*****
  111 IB=1
      IF (TYPE.EQ.1HA) GOTO 120
      IF (TYPE.EQ.1HF) GOTO 130
      DECODE (80,3,TEMP1(3)) ((IYR(I),VAL(I)),I=1,6)
      DO 112 I=1,6
      IF (IYR(I).EQ.10H          )GOTO 1000
      J=IYR(I)-IYR1+1
      BDATA(J)=VAL(I)
  112 CONTINUE
      GOTO 1000
  120 DECODE (80,4,TEMP1(3)) VALU
      DO 125 I=1,NTOTYR
      BDATA(I)=VALU
  125 CONTINUE
      GOTO 1000
  130 DECODE (80,5,TEMP1(3)) NSTART,NEND,VALU
      I1=NSTART-IYR1+1
      I2=NEND-IYR1+1
      DO 135 I=I1,I2
      BDATA(I)=VALU
  135 CONTINUE
      GOTO 1000
C*****
C       THIS SECTION PROCESSES C TYPE DATA CARDS
C*****
  138 IC=1
      DECODE (80,11,TEMP1(3)) (IFN(I),I=1,5)
      IF (CARDNO.NE.1H1) GOTO 141
      I=IFN(1)
      DO 140 J=1,NTOTYR
      CDATA(J)=FUNC(I,J)
  140 CONTINUE
      KL=2
      GOTO 142
  141 KL=1
  142 DO 145 I=KL,5
      IF (IFN(I).EQ.LANK) GOTO 1000
      J=IFN(I)
      DO 143 K=1,NTOTYR
      CDATA(K)=CDATA(K)*FUNC(J,K)
  143 CONTINUE
  145 CONTINUE
  150 GOTO 1000
C*****
C       THIS SECTION PROCESSES D DATA CARDS
C*****
  160 ID=1
      IF (CARDNO.NE.1H1) GOTO 163
      DO 161 I=1,NTOTYR
      DDATA(I)=0.
  161 CONTINUE
  163 IF (TYPE.EQ.1HA) GOTO 170
      IF (TYPE.EQ.1HF) GOTO 180
      DECODE (80,3,TEMP1(3)) ((IYR(I),VAL(I)),I=1,6)
      DO 162 I=1,6
      IF (IYR(I).EQ.2H  ) GOTO 1000
      J=IYR(I)-IYR1+1
      DDATA(J)=VAL(I)
  162 CONTINUE
      GOTO 1000
  170 DECODE (80,4,TEMP1(3)) VALU
      DO 172 I=1,NTOTYR
      DDATA(I)=VALU
  172 CONTINUE
      GOTO 1000
  180 DECODE (80,5,TEMP1(3)) NSTART,NEND,VALU
      I1=NSTART-IYR1+1
      I2=NEND-IYR1+1
      DO 182 I=I1,I2
      DDATA(I)=VALU
  182 CONTINUE
      GOTO 1000
C*****
C       THIS SECTION PROCESSES E DATA CARDS
C*****
```

```
  183 IE=1
      IF (CARDNO.NE.1H1) GOTO 184
      DO 186 I=1,NTOTYR
      EDATA(I)=0.
  186 CONTINUE
  184 DECODE (80,11,TEMP1(3)) (IFN(I),I=1,5)
      I=IFN(1)
      DO 185 J=1,NTOTYR
      EDATA(J)=FUNC(I,J)
  185 CONTINUE
      DO 190 I=2,5
      IF (IFN(I).EQ.2H  ) GOTO 1000
      J=IFN(I)
      DO 188 K=1,NTOTYR
      EDATA(K)=EDATA(K)+FUNC(J,K)
  188 CONTINUE
  190 CONTINUE
      GOTO 1000
C*****
C          THIS SECTION PROCESSES G AND H DATA CARDS
C*****
  200 IG=1
      DECODE (80,12,TEMP1(3)) A,B,IQTY1,IFN
      READ (1,1) (TEMP2(I),I=1,8)
      DECODE (80,13,TEMP2(3)) ((NUNIT(I),QTY(I)),I=1,5)
      DO 210 I=1,5
      IF (NUNIT(I).EQ.10H          ) GOTO 211
  210 CONTINUE
      NALL=5
      NUNIT(6)=10000000000
      GOTO 213
  211 NUNIT(I)=10000000000
  213 CONTINUE
      NTOT1=IQTY1
      DO 215 I=1,5
      J=I+1
      IF (NTOT1.LT.NUNIT(J)) GOTO 218
  215 CONTINUE
  218 ICL=I
      DO 290 IYR=1,NTOTYR
      TRACK=0.
      IQUAN=FUNC(IFN,IYR)
      NTOT2=NTOT1+IQUAN
      IF (NTOT2.EQ.NTOT1) GOTO 290
      IF (NTOT2.GT.NUNIT(ICL+1)) GOTO 230
      IF (IQUAN.LE.30) GOTO 225
      CALL WEDDLE(A,B,NTOT1,NTOT2,QTY(ICL),TOTAL)
      TRACK=TRACK+TOTAL
      GDATA(IYR)=TRACK
      NTOT1=NTOT2
      GOTO 280
  225 NBOOB=NTOT1
      CALL SUMUP(A,B,NBOOB,NTOT2,QTY(ICL),TOTAL)
      TRACK=TRACK+TOTAL
      GDATA(IYR)=TRACK
      GOTO 280
  230 DO 250 JJ=ICL,5
      IF (NTOT2.LE.NUNIT(JJ+1)) GOTO 260
      NUMB=NUNIT(JJ+1)-NTOT1
      IF (NUMB.GT.30) GOTO 235
      NBOOB=NTOT1
      CALL SUMUP (A,B,NBOOB,NUNIT(JJ+1),QTY(JJ),TOTAL)
      TRACK=TRACK+TOTAL
      GOTO 240
  235 CALL WEDDLE (A,B,NTOT1,NUNIT(JJ+1),QTY(JJ),TOTAL)
      TRACK=TRACK+TOTAL
  240 NTOT1=NTOT1+NUMB+1
  250 CONTINUE
  260 NUMB=NTOT2-NTOT1
      IF (NUMB.GT.30) GOTO 265
      NBOOB=NTOT1
      CALL SUMUP (A,B,NBOOB,NTOT2,QTY(JJ),TOTAL)
      ICL=JJ
      GOTO 270
  265 CALL WEDDLE (A,B,NTOT1,NTOT2,QTY(JJ),TOTAL)
      ICL=JJ
  270 TRACK=TRACK+TOTAL
      GDATA(IYR)=TRACK
  280 NTOT1=NTOT2+1
  290 CONTINUE
  300 CONTINUE
      GOTO 1000
C*****
C         AT THIS POINT WE HAVE PROCESSED A CARD,
C         IF THE NEXT DATA CARD IS OF THE SAME LEVEL WE WILL PROCESS IT,
C         IF NOT WE CAN DETERMINE ITS DEGREE, PERFORM THE CALCULATIONS
C         ON THE PREVIOUS LEVEL, AND STORE THE RESULT ACCORDING TO HOW
C         THE LEVEL DEGREES COMPARE
C*****
 1000 READ(1,1) (TEMP2(I),I=1,8)
      DECODE (80,9,TEMP2(1)) (LEVEL2(I),I=1,18)
      DO 1005 I=1,18
      IF (LEVEL1(I).NE.LEVEL2(I)) GOTO 1010
 1005 CONTINUE
      DO 1006 I=1,8
      TEMP1(I)=TEMP2(I)
 1006 CONTINUE
      GOTO 102
C*****
C     IF HERE WE GOTO 102 WE ARE STILL PROCESSING DATA FROM THIS LEVEL
C     IF WE GO TO 1010 WE HAVE COMPLETED PROCESSING CARDS FROM
C     THIS LEVEL AND MUST CALCULATE THE COSTS OF THIS LEVEL
C*****
 1010 CALL DEGREE (LEVEL2,LEV2,1,18)
      IF (IB.EQ.0.AND.IC.EQ.0.AND.ID.EQ.0.
     1  AND.IE.EQ.0.AND.IG.EQ.0) GO TO 1031
      IF(IA.EQ.1.AND.IB.EQ.1.AND.IC.EQ.0.AND.ID.EQ.0.AND.IE.EQ.0.
     1   AND.IG.EQ.0) GOTO 1034
      DO 1011 I=1,NTOTYR
      FINDAT(I)=1.
 1011 CONTINUE
      IF (IB.EQ.0) GOTO 1015
      DO 1012 I=1,NTOTYR
      FINDAT(I)=FINDAT(I)*BDATA(I)
 1012 CONTINUE
 1015 IF (IC.EQ.0) GOTO 1020
      DO 1017 I=1,NTOTYR
      FINDAT(I)=FINDAT(I)*CDATA(I)
 1017 CONTINUE
 1020 IF (ID.EQ.0) GOTO 1025
```

```
      DO 1022 I=1,NTOTYR
      FUNC(98,I)=DDATA(I)
      FINDAT(I)=FINDAT(I)*DDATA(I)
 1022 CONTINUE
 1025 IF (IE.EQ.0) GOTO 1030
      DO 1027 I=1,NTOTYR
      FUNC(99,I)=EDATA(I)
      FINDAT(I)=FINDAT(I)*EDATA(I)
 1027 CONTINUE
 1030 IF (IG.EQ.0) GOTO 1035
      DO 1032 I=1,NTOTYR
      FINDAT(I)=FINDAT(I)*GDATA(I)
 1032 CONTINUE
      GOTO 1035
 1031 DO 1033 I=1,NTOTYR
      FINDAT(I)=0.
 1033 CONTINUE
      GOTO 1035
 1034 DO 1037 I=1,NTOTYR
      FINDAT(I)=BDATA(I)*1000000.
 1037 CONTINUE
 1035 DO 1036 I=1,NTOTYR
      FINDAT(I)=FINDAT(I)/1000000.
 1036 CONTINUE
      IA=0
      IB=0
      IC=0
      ID=0
      IE=0
      IG=0
C*****
C         AT THIS POINT FINDAT ARRAY CONTAINS THE
C         DATA FOR THE LEVEL WE HAVE JUST PROCESSED;
C         MUST NOW DETERMINE HOW THIS LEVEL'S DATA
C         SHOULD BE STORED BY COMPARISON WITH THE
C         DEGREE OF THE NEXT LEVEL.
C*****
      DO 1038 I=1,NTOTYR
      BDATA(I)=0.
      CDATA(I)=0.
      DDATA(I)=0.
      EDATA(I)=0.
      GDATA(I)=0.
 1038 CONTINUE
      IF (LEV2.EQ.0) KDONE=1
      IF (ISTORE(LEV1,3).EQ.1) GO TO 1200
      KTAPE=KTAPE+1
      KOP=9-LEV1+1
      DO 1130 I=1,KOP
      J=9-I+1
      IF (ISTORE(J,3).EQ.1) GO TO 1130
      DO 1126 K=1,NTOTYR
      STORE(J-1,K)=STORE(J-1,K)+STORE(J,K)
 1126 CONTINUE
      WRITE (2,18) ((ISTORE(J,K),K=1,2),(ISTORE(J,K),K=4,12))
      WRITE (2,19) (STORE(J,K),K=1,NTOTYR)
      ISTORE(J,3)=1
 1130 CONTINUE
 1200 CALL PACK(LEVEL1(1),IZ,18)
      ISTORE(LEV1,1)=IZ(1)
      ISTORE(LEV1,2)=IZ(2)
      ISTORE(LEV1,3)=0
      DO 1205 I=1,12
      IJK=I+3
      ISTORE(LEV1,IJK)=INAME1(I)
      INAME1(I)=LANK
 1205 CONTINUE
      DO 1208 K=1,NTOTYR
      STORE(LEV1,K)=FINDAT(K)
 1208 CONTINUE
      IF(KDONE.EQ.1) GOTO 1320
      IF(LEV1.GE.LEV2) GOTO 1220
      DO 1210 I=1,8
      TEMP1(I)=TEMP2(I)
 1210 CONTINUE
      LEV1=LEV2
      GOTO 102
 1220 IF(LEV1.GT.LEV2) GOTO 1250
      KTAPE=KTAPE+1
      WRITE (2,18) ((ISTORE(LEV1,K),K=1,2),(ISTORE(LEV1,K),K=4,12))
      WRITE (2,19) (STORE(LEV1,K),K=1,NTOTYR)
      ISTORE(LEV1,3)=1
      MINUS=LEV1-1
      DO 1230 I=1,NTOTYR
      STORE(MINUS,I)=STORE(MINUS,I)+STORE(LEV1,I)
 1230 CONTINUE
      DO 1240 I=1,8
      TEMP1(I)=TEMP2(I)
 1240 CONTINUE
      LEV1=LEV2
      GOTO 102
 1250 NDIF=LEV1-LEV2
      J=LEV1
      DO 1270 KKK=1,NDIF
      KTAPE=KTAPE+1
      WRITE (2,18) ((ISTORE(J,K),K=1,2),(ISTORE(J,K),K=4,12))
      WRITE (2,19) (STORE(J,K),K=1,NTOTYR)
      ISTORE(J,3)=1
      DO 1260 K=1,NTOTYR
      STORE(J-1,K)=STORE(J-1,K)+STORE(J,K)
 1260 CONTINUE
      J=J-1
 1270 CONTINUE
      IF(LEV2.EQ.1) GOTO 1300
      DO 1280 K=1,NTOTYR
      STORE(LEV2-1,K)=STORE(LEV2-1,K)+STORE(LEV2,K)
 1280 CONTINUE
 1300 KTAPE=KTAPE+1
      WRITE (2,18) ((ISTORE(LEV2,K),K=1,2),(ISTORE(LEV2,K),K=4,12))
      WRITE (2,19) (STORE(LEV2,K),K=1,NTOTYR)
      ISTORE(LEV2,3)=1
      DO 1310 I=1,8
      TEMP1(I)=TEMP2(I)
 1310 CONTINUE
      LEV1=LEV2
      GOTO 102
 1320 J=LEV1
      DO 1340 I=1,LEV1
      KTAPE=KTAPE+1
      WRITE (2,18) ((ISTORE(J,K),K=1,2),(ISTORE(J,K),K=4,12))
      WRITE (2,19) (STORE(J,K),K=1,NTOTYR)
      IF (J-1.EQ.0) GO TO 1340
      DO 1330 K=1,NTOTYR
      STORE(J-1,K)=STORE(J-1,K)+STORE(J,K)
 1330 CONTINUE
      ISTORE(J,3)=1
      J=J-1
 1340 CONTINUE
      WRITE (2,17) XER
      END FILE 2
      STOP
      END
      SUBROUTINE WEDDLE (A,B,I1,I2,FACTOR,TOTAL)
      BOT=1.-B
      CON=A/BOT
      FIRST=(FLOAT(I2))**(1.-B)
      SECOND=(FLOAT(I1))**(1.-B)
      FORCE=FIRST-SECOND
      TOTAL=FORCE*CON*FACTOR
      RETURN
      END
      SUBROUTINE SUMUP (A,B,I1,I2,FACTOR,TOTAL)
      TOTAL=0.
      IF(I1.EQ.0) I1=1
      DO 10 I=I1,I2
      XI=I
      TOTAL=TOTAL+(A*XI**(-B))
   10 CONTINUE
      TOTAL=TOTAL*FACTOR
      RETURN
      END
      SUBROUTINE DEGREE (IARRAY,NDEG,ISTART,ISTOP)
      DIMENSION IARRAY(25)
      DO 10 I=ISTART,ISTOP
      IF (IARRAY(I).EQ.1H ) GO TO 15
   10 CONTINUE
      NDEG=9
      GO TO 20
   15 NDEG=(I-1)/2
   20 CONTINUE
      RETURN
      END
*     LIST(STOP)
```

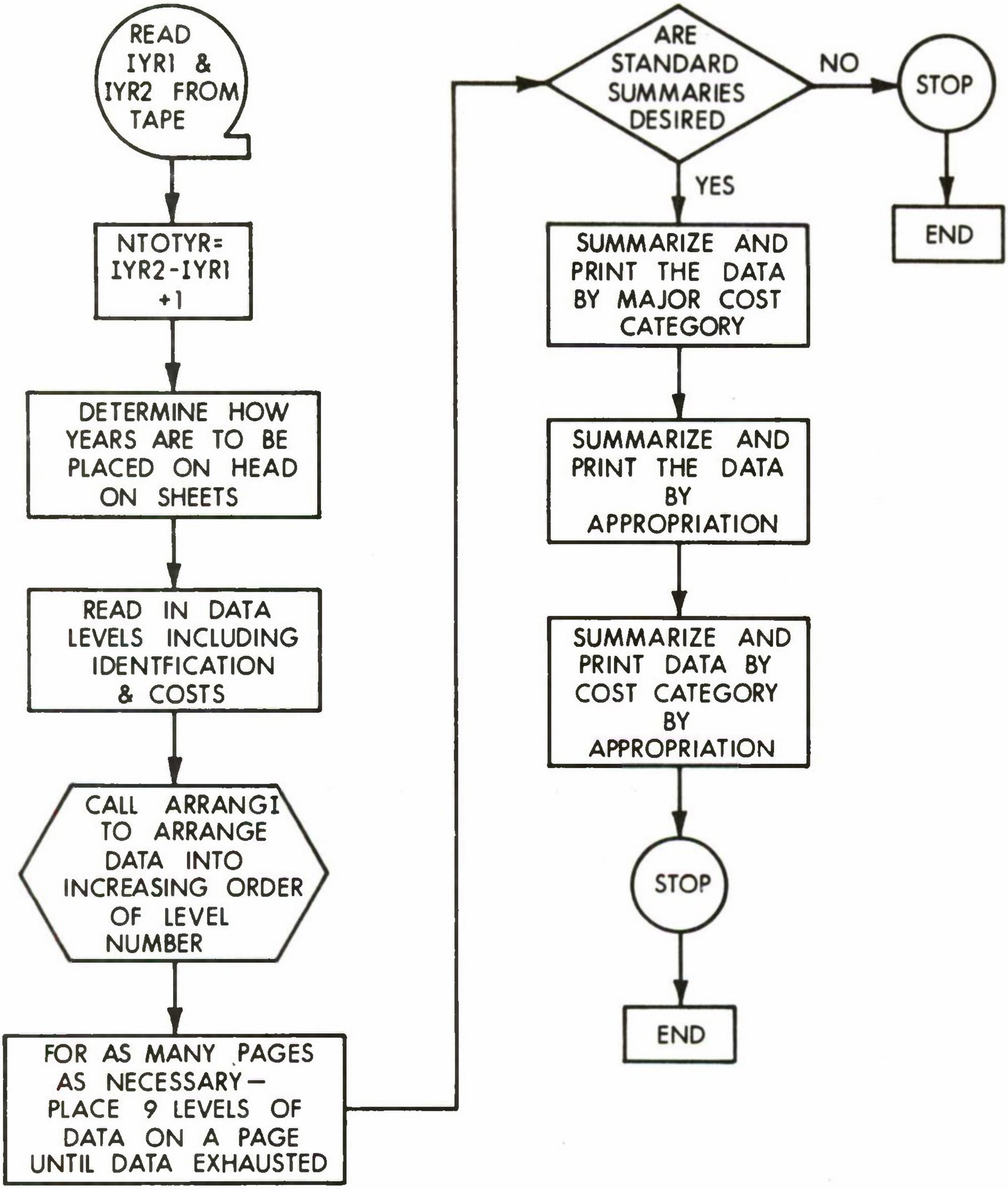

READ IYR1 & IYR2 FROM TAPE
NTOTYR= IYR2-IYR1 +1
DETERMINE HOW YEARS ARE TO BE PLACED ON HEAD ON SHEETS
READ IN DATA LEVELS INCLUDING IDENTFICATION & COSTS
CALL ARRANGI TO ARRANGE DATA INTO INCREASING ORDER OF LEVEL NUMBER
FOR AS MANY PAGES AS NECESSARY – PLACE 9 LEVELS OF DATA ON A PAGE UNTIL DATA EXHAUSTED
ARE STANDARD SUMMARIES DESIRED
NO
STOP
END
YES
SUMMARIZE AND PRINT THE DATA BY MAJOR COST CATEGORY
SUMMARIZE AND PRINT THE DATA BY APPROPRIATION
SUMMARIZE AND PRINT DATA BY COST CATEGORY BY APPROPRIATION
STOP
END

Figure A-4 Flow Chart – Report Preparation and Summarization Program

A listing of the Report Preparation and Summarization Program (Program 3 in the sequence) follows.

```
 FEB. 7,72  BRLESC2 FORTRAN.                                         CB 17..32
*       SA11WC 392 107 4210 COPES  PROGRAM 3

$       COMMENT TAPE UNIT 2 USED FOR INPUT
        DIMENSION ID(9),IDATE(3),NCOUNT(3),NYR(30)
        DIMENSION VALUE(100,43),IT(25),INT(25)
        DIMENSION TEMPO(4,7,30),TOTAL(4,7),COSCAT(30),APPRO(30)
        DIMENSION FMT(6),GMT(4),HMT(4),FMT1(3),GMT1(3),HMT1(5)
      1 FORMAT(I4,2X,I4)
      2 FORMAT(5A10,10X,I2,1X,A9,1X,I4)
      3 FORMAT (A10,A8,A9,5A10,A9,2A10)
      4 FORMAT (10F12.3,/,10F12.3,/,10F12.3)
      5 FORMAT(1H1,120X,'PAGE',1X,I3)
      6 FORMAT (A9,5A10,A9,2A10, 5X,I2,1X,A9,1X,I4)
      7 FORMAT (40X,'COST DATA IN MILLIONS')
     10 FORMAT(1H0,A9,5A10,55X,A10,A8)
     11 FORMAT (1H0,A9,5A10,/,A9,2A10,85X,A10,A8)
     14 FORMAT(1H0,120X,'PAGE',1X,I3)
     15 FORMAT (8F10.3)
     16 FORMAT (2X,I2,I6)
     17 FORMAT (1H0,2X,'DEVELOPMENT')
     18 FORMAT (1H0,2X,'INVESTMENT-NON RECURRING')
     19 FORMAT (1H0,2X,'INVESTMENT RECURRING')
     20 FORMAT (1H0,2X,'OPERATING')
     21 FORMAT (1H0,'SUMMARY BY APPROPRIATION')
     22 FORMAT (I2,I6)
     23 FORMAT (1H0,2X,'RDT-E')
     24 FORMAT (1H0,2X,'PEMA')
     25 FORMAT (1H0,2X,'O-MA')
     26 FORMAT (1H0,2X,'MPA')
     27 FORMAT (1H0,2X,'MCA')
     28 FORMAT (1H0,2X,'ASF')
     29 FORMAT (1H0,2X,'FHMA')
     30 FORMAT (1H0,'SUMMARY BY MAJOR COST CATEGORY')
     31 FORMAT (2X,I2)
     32 FORMAT (1H0,'SUMMARY-COST CATEGORY BY APPROPRIATION')
     33 FORMAT (I2)
     34 FORMAT (10A1)
     35 FORMAT (8A1)
     36 FORMAT (I10)
     37 FORMAT (2X,I2)
     38 FORMAT (I2,I6)
        INTEGER FMT,FMT1,GMT,GMT1,HMT,HMT1
        DATA (FMT(I),I=1,6) /8H(1H ,2X,,1H ,10H('FY',I2,6,3HX),,1H ,
      1  10HX,'TOTAL')/
        DATA (GMT(I),I=1,4) /8H(1H ,3X,,1H ,10H('FY',I2,6,3HX))/
        DATA (HMT(I),I=1,4) /8H(1H ,4X,,1H ,10H('FY',I2,6,3HX))/
        DATA (FMT1(I),I=1,3) /8H(1H ,2X,,1H ,10H(F9.3,2X))/
        DATA (GMT1(I),I=1,3) /8H(1H ,3X,,1H ,10H(F9.3,2X))/
        DATA (HMT1(I),I=1,5) /8H(1H ,4X,,1H ,10H(F9.3,2X),,1H ,8HX,F10.3)/
        REWIND 2
        READ (2,6) ((ID(I),I=1,9),(IDATE(I),I=1,3))
        READ (2,1) IYR1,IYREND
        NTOTYR=IYREND-IYR1+1
        NMOD=MOD(NTOTYR,3)
        IF(NMOD.EQ.0)GOTO 110
        IF(NMOD.EQ.1)GOTO 120
        NCOUNT(1)=(NTOTYR/3)+1
        NCOUNT(2)=NCOUNT(1)
        NCOUNT(3)=NCOUNT(1)-1
      GOTO 140
  110 NOD=(NTOTYR/3)
      DO 112 I=1,3
      NCOUNT(I)=NOD
  112 CONTINUE
      GOTO 140
  120 NCOUNT(1)=(NTOTYR/3)+1
      NCOUNT(2)=NCOUNT(1)-1
      NCOUNT(3)=NCOUNT(2)
  140 LOBO=NCOUNT(1)
      MOBO=NCOUNT(2)
      NOBO=NCOUNT(3)
      IF(LOBO.EQ.10) GOTO 141
      FMT(2)=LOBO
      FMT1(2)=LOBO
      GOTO 142
  141 FMT(2)=112
      FMT1(2)=112
  142 IF(MOBO.EQ.10) GOTO 143
      GMT(2)=MOBO
      GMT1(2)=MOBO
      GOTO 144
  143 GMT(2)=112
      GMT1(2)=112
  144 IF(NOBO.EQ.10) GOTO 146
      HMT(2)=NOBO
      HMT1(2)=NOBO
      GOTO 147
  146 HMT(2)=112
      HMT1(2)=112
  147 DO 145 I=1,LOBO
      NYR(I)=IYR1+(3*(I-1))
  145 CONTINUE
      L2=LOBO+1
      L3=L2+NCOUNT(2)-1
      J=1
      DO 148 I=L2,L3
      NYR(I)=(IYR1+1)+(J-1)*3
      J=J+1
  148 CONTINUE
      J=1
      L4=L3+1
      L5=L4+NOBO-1
      DO 150 I=L4,L5
      NYR(I)=(IYR1+2)+(J-1)*3
      J=J+1
  150 CONTINUE
      DECODE(10,36,FMT(2)) NFMT
      NSPACF=117-2-NFMT*10
      ENCODE(10,1,FMT(5)) NSPACF
      DECODE(10,36,HMT1(2)) MFMT
      MSPACF=117-4-MFMT*11
      ENCODE(10,1,HMT1(4)) MSPACF
      NFINAL=11+NTOTYR
      DO 200 I=1,1000
      READ (2,3) (VALUE(I,J),J=1,11)
      IF(VALUE(I,1).EQ.5HXXXXX)GOTO 210
      READ (2,4) (VALUE(I,J),J=12,NFINAL)
  200 CONTINUE
  210 NUM=I-1
```

```
      DO 212 J=1,NUM
      DECODE(80,34,VALUE(J,1)) (IT(K),K=1,10)
      DECODE(80,35,VALUE(J,2)) (IT(K),K=11,18)
      DO 211 K=1,18
      INT(K)=IT(K)
      IF (IT(K).NE.1H0) GOTO 211
      INT(K)=0
  211 CONTINUE
      ENCODE(80,34,VALUE(J,42)) (INT(K),K=1,10)
      ENCODE(80,35,VALUE(J,43)) (INT(K),K=11,18)
  212 CONTINUE
      CALL ARANGI(VALUE,NUM,43,100,43)
      CALL ARANGI(VALUE,NUM,42,100,43)
C****
C           NUM CONTAINS TOTAL NUMBER OF LEVELS USED
C           VALUE NOW CONTAINS THE ORDERED LEVELS AND DATA
C****
      INUM=1
      DO 300 IPAGE=1,500
      PRINT 5,IPAGE
      PRINT 6,((ID(I),I=1,9),(IDATE(I),I=1,3))
      PRINT 7
      PRINT FMT , (NYR(I),I=1,LOBO)
      PRINT GMT , (NYR(I),I=L2,L3)
      PRINT HMT , (NYR(I),I=L4,L5)
      INUM9=INUM+9
      DO 250 I=INUM,INUM9
      DO 215 J=9,11
      IF(VALUE(I,J).NE.1H ) GOTO 225
  215 CONTINUE
      PRINT 10,((VALUE(I,J),J=3,8),(VALUE(I,J),J=1,2))
      GOTO 230
  225 PRINT 11, ((VALUE(I,J),J=3,11),(VALUE(I,J),J=1,2))
  230 TO=0.
      DO 231 KO=12,NFINAL
      TO=TO+VALUE(I,KO)
  231 CONTINUE
      PRINT FMT1,(VALUE(I,K),K=12,NFINAL,3)
      PRINT GMT1,(VALUE(I,K),K=13,NFINAL,3)
      PRINT HMT1,((VALUE(I,K),K=14,NFINAL,3),TO)
      IF(I.EQ.NUM)GOTO 310
  250 CONTINUE
      PRINT 14,IPAGE
      INUM=INUM9+1
  300 CONTINUE
  310 CONTINUE
      READ 16, NOY
      IF(NOY.EQ.0) GOTO 410
C*****
C         CALCULATION OF SUMMARY 1
C                       BY COST CATEGORY
C*****
      IPAGE=IPAGE+1
      PRINT 5, IPAGE
      PRINT 6,((ID(I),I=1,9),(IDATE(I),I=1,3))
      PRINT FMT , (NYR(I),I=1,LOBO)
      PRINT GMT , (NYR(I),I=L2,L3)
      PRINT HMT , (NYR(I),I=L4,L5)
      J=1
      I=1
  312 IF (J.EQ.5) GOTO 325
      DECODE (80,16,VALUE(I,1)) ITEST1,ITEST2
      IF (ITEST1.NE.J.OR.ITEST2.NE.0) GOTO 320
      IER=1
      DO 315 K=12,NFINAL
      TEMPO(J,1,IER)=TEMPO(J,1,IER)+VALUE(I,K)
      IER=IER+1
  315 CONTINUE
      J=J+1
      I=I+1
      GOTO 312
  320 I=I+1
      GOTO 312
C*****
C         TEMPO ARRAY CONTAINS DATA FOR 1ST SUMMARY
C            WILL CALCULATE TOTAL ARRAY AND TOTAL BY
C               YEAR, THEN PRINT THE RESULTS.
C*****
  325 DO 340 I=1,4
      DO 335 K=1,NTOTYR
      TOTAL(I,1)=TOTAL(I,1)+TEMPO(I,1,K)
      COSCAT(K)=COSCAT(K)+TEMPO(I,1,K)
  335 CONTINUE
  340 CONTINUE
      PRINT 30,
      DO 341 I=1,NTOTYR
      COSTOT=COSTOT+COSCAT(I)
  341 CONTINUE
      PRINT FMT1,(COSCAT(K),K=1,NTOTYR,3)
      PRINT GMT1,(COSCAT(K),K=2,NTOTYR,3)
      PRINT HMT1,(COSCAT(K),K=3,NTOTYR,3),COSTOT
      DO 350 J=1,4
      GOTO (342,343,344,345),J
  342 PRINT 17,
      GOTO 346
  343 PRINT 18,
      GOTO 346
  344 PRINT 19,
      GOTO 346
  345 PRINT 20,
  346 PRINT FMT1,(TEMPO(J,1,K),K=1,NTOTYR,3)
      PRINT GMT1,(TEMPO(J,1,K),K=2,NTOTYR,3)
      PRINT HMT1,(TEMPO(J,1,K),K=3,NTOTYR,3),TOTAL(J,1)
  350 CONTINUE
C*****
C            CALCULATION OF SUMMARY 2
C                BY APPROPRIATION
C*****
      DO 348 I=1,4
      DO 348 J=1,7
      TOTAL(I,J)=0.
      DO 348 K=1,30
      TEMPO(I,J,K)=0.
      COSCAT(K)=0.
  348 CONTINUE
      COSTOT=0.
      DO 370 L=1,NUM
      DECODE (80,22,VALUE(L,2)) ITEST1,ITEST2
       DO 355 J=1,7
      IF (ITEST1.EQ.J.AND.ITEST2.EQ.0) GOTO 360
  355 CONTINUE
      GOTO 370
  360 IER=1
      DO 361 K=12,NFINAL
      APPRO(IER)=APPRO(IER)+VALUE(L,K)
      TEMPO(1,J,IER)=TEMPO(1,J,IER)+VALUE(L,K)
      IER=IER+1
  361 CONTINUE
  370 CONTINUE
      DO 390 J=1,7
      DO 390 K=1,NTOTYR
      TOTAL(1,J)=TOTAL(1,J)+TEMPO(1,J,K)
  390 CONTINUE
      DO 391 K=1,NTOTYR
      APTOT=APTOT+APPRO(K)
  391 CONTINUE
      IF(APTOT.EQ.0.) GOTO 411
      IPAGE=IPAGE+1
      PRINT 5, IPAGE
      PRINT 6,((ID(I),I=1,9),(IDATE(I),I=1,3))
      PRINT FMT , (NYR(I),I=1,LOBO)
      PRINT GMT , (NYR(I),I=L2,L3)
      PRINT HMT , (NYR(I),I=L4,L5)
      PRINT 21,
      PRINT FMT1,(APPRO(K),K=1,NTOTYR,3)
      PRINT GMT1,(APPRO(K),K=2,NTOTYR,3)
      PRINT HMT1,(APPRO(K),K=3,NTOTYR,3),APTOT
      DO 410 J=1,7
      IF(TOTAL(1,J).EQ.0.) GOTO 410
      GOTO (402,403,404,405,406,407,408),J
  402 PRINT 23,
      GOTO 409
  403 PRINT 24,
      GOTO 409
  404 PRINT 25,
      GOTO 409
  405 PRINT 26,
      GOTO 409
  406 PRINT 27,
      GOTO 409
  407 PRINT 28,
      GOTO 409
  408 PRINT 29,
  409 PRINT FMT1,(TEMPO(1,J,K),K=1,NTOTYR,3)
      PRINT GMT1,(TEMPO(1,J,K),K=2,NTOTYR,3)
      PRINT HMT1,(TEMPO(1,J,K),K=3,NTOTYR,3),TOTAL(1,J)
  410 CONTINUE
C*****
C           CALCULATION OF SUMMARY 3
C                   COST  CATEGORY BY APPROPRIATION
C*****
  411 IPAGE=IPAGE+1
      PRINT 5, IPAGE
      PRINT 6, ((ID(I),I=1,9),(IDATE(I),I=1,3))
      DO 420 I=1,4
      DO 420 J=1,7
      TOTAL(I,J)=0.
      DO 420 K=1,NTOTYR
      TEMPO(I,J,K)=0.
      COSCAT(K)=0.
  420 CONTINUE
      COSTOT=0.
      PRINT FMT, (NYR(I),I=1,LOBO)
      PRINT GMT, (NYR(I),I=L2,L3)
      PRINT HMT, (NYR(I),I=L4,L5)
      DO 430 I=1,4
      DO 425 J=1,7
      DO 424 L=1,NUM
      DECODE (80,37,VALUE(L,1)) ITEST1
      DECODE (80,38,VALUE(L,2)) ITEST2,ITEST3
      IF(ITEST1.NE.I.OR.ITEST2.NE.J.OR.ITEST3.NE.0) GOTO 424
      IER=1
      DO 412 K=12,NFINAL
      TEMPO(I,J,IER)=TEMPO(I,J,IER)+VALUE(L,K)
      IER=IER+1
  412 CONTINUE
  424 CONTINUE
      DO 421 K=1,NTOTYR
      TOTAL(I,J)=TOTAL(I,J)+TEMPO(I,J,K)
      COSCAT(K)=COSCAT(K)+TEMPO(I,J,K)
  421 CONTINUE
  425 CONTINUE
  430 CONTINUE
      DO 422 K=1,NTOTYR
      COSTOT=COSTOT+COSCAT(K)
  422 CONTINUE
      IF (COSTOT.EQ.0.) STOP
      PRINT 32,
      PRINT FMT1, (COSCAT(K),K=1,NTOTYR,3)
      PRINT GMT1, (COSCAT(K),K=2,NTOTYR,3)
      PRINT HMT1, (COSCAT(K),K=3,NTOTYR,3),COSTOT
      DO 500 I=1,4
      IPRINT=0
      DO 480 J=1,7
      IF (TOTAL(I,J).EQ.0.) GOTO 480
      IF(IPRINT.EQ.1) GOTO 460
      GOTO (456,457,458,459),I
  456 PRINT 17,
      IPRINT =1
      GOTO 460
  457 PRINT 18,
      IPRINT =1
      GOTO 460
  458 PRINT 19,
      IPRINT =1
      GOTO 460
  459 PRINT 20,
      IPRINT =1
  460 GOTO (461,462,463,464,465,466,467),J
  461 PRINT 23,
      GOTO 470
  462 PRINT 24,
      GOTO 470
  463 PRINT 25,
      GOTO 470
  464 PRINT 26,
      GOTO 470
  465 PRINT 27,
      GOTO 470
  466 PRINT 28,
      GOTO 470
  467 PRINT 29,
  470 PRINT FMT1, (TEMPO(I,J,K),K=1,NTOTYR,3)
      PRINT GMT1, (TEMPO(I,J,K),K=2,NTOTYR,3)
      PRINT HMT1, (TEMPO(I,J,K),K=3,NTOTYR,3),TOTAL(I,J)
  480 CONTINUE
  500 CONTINUE
      STOP
      END
      SUBROUTINE ARANGI(IA,N,L,NMAX,LMAX)
      DIMENSION IA(NMAX,LMAX)
C          ORDERS LTH ELEMENTS OF ARRAY IA FROM SMALLEST TO LARGEST VALUE
C          THE OTHER (LMAX-1) ELEMENTS OF ARRAY IA ARE CARRIED ALONG
C            ONE TO ONE AS THE LTH ELEMENT IS SEQUENCED
      IF(N.LE.1)RETURN
      DO 40 I2=2,N $ I1=I2-1
      IF(IA(I1,L).LE.IA(I2,L))GOTO 40
      DO 10 LL=1,LMAX
      IT=IA(I1,LL) $ IA(I1,LL)=IA(I2,LL) $ IA(I2,LL)=IT
   10 CONTINUE
      J2=I1
   20 IF(J2.LE.1)GOTO 40 $ J1=J2-1
      IF(IA(J1,L).LE.IA(J2,L))GOTO 40
      DO 30 LL=1,LMAX
      IT=IA(J1,LL) $ IA(J1,LL)=IA(J2,LL) $ IA(J2,LL)=IT
   30 CONTINUE $ J2=J2-1 $ GOTO 20
   40 CONTINUE $ RETURN $ END
*     LIST(STOP)
```

DISTRIBUTION LIST

No. of Copies — Organization

12 Commander
Defense Documentation Center
ATTN: TIPCR
Cameron Station
Alexandria, Virginia 22314

1 Commanding General
U.S. Army Materiel Command
ATTN: AMCDL
Washington, D.C. 20315

1 Commanding General
U.S. Army Materiel Command
ATTN: AMCMA
Washington, D.C. 20315

1 Commanding General
U.S. Army Materiel Command
ATTN: AMCMS
Washington, D.C. 20315

1 Commanding General
U.S. Army Materiel Command
ATTN: AMCQA
Washington, D.C. 20315

1 Commanding General
U.S. Army Materiel Command
ATTN: AMCRD
Washington, D.C. 20315

1 Commanding General
U.S. Army Materiel Command
ATTN: AMCRD-E
Washington, D.C. 20315

1 Commanding General
U.S. Army Materiel Command
ATTN: AMCRD-G
Washington, D.C. 20315

1 Commanding General
U.S. Army Materiel Command
ATTN: AMCRD-M
Washington, D.C. 20315

No. of Copies — Organization

1 Commanding General
U.S. Army Materiel Command
ATTN: AMCRD-R
Washington, D.C. 20315

1 Commanding General
U.S. Army Materiel Command
ATTN: AMCRD-P
Washington, D.C. 20315

1 Commanding General
U.S. Army Materiel Command
ATTN: AMCRD-W
Washington, D.C. 20315

1 Commanding General
U.S. Army Materiel Command
ATTN: AMCRD-T
Washington, D.C. 20315

1 Commanding General
U.S. Army Materiel Command
ATTN: AMCSA-PM-MBT
Washington, D.C. 20315

1 Commanding General
U.S. Army Materiel Command
ATTN: AMCCP
Washington, D.C. 20315

1 Commanding General
U.S. Army Materiel Command
ATTN: AMCPA-SA
Washington, D.C. 20315

1 Commanding General
U.S. Army Materiel Command
ATTN: AMCPM-SA
Picatinny Arsenal
Dover, New Jersey 07801

1 Director
MUCOM Operations Research Group
Edgewood Arsenal, Maryland
21010

DISTRIBUTION LIST

No. of Copies — Organization

1 Commanding Officer
U.S. Army Harry Diamond Laboratories
ATTN: Sys Anal Ofc
Washington, D.C. 20438

1 Commanding Officer
Frankford Arsenal
ATTN: Mr. George Schecter
Philadelphia, Pennsylvania
19137

1 Chief, Analytical Sciences Office
USA Biological Defense Research Laboratory
ATTN: AMXBL-AS
Dugway, Utah 84022

1 Commanding General
U.S. Army Aviation Systems Command
ATTN: AMSAV-R-X (Mr. Hollis)
P.O. Box 209
St. Louis, Missouri 63166

1 Commanding General
U.S. Army Electronics Command
ATTN: AMSEL-PL
Fort Monmouth, New Jersey
07703

1 Commanding General
U.S. Army Missile Command
ATTN: AMSMI-DA
Redstone Arsenal, Alabama
35809

No. of Copies — Organization

1 Commanding General
U.S. Army Mobility Equipment Command
ATTN: Sys Anal Ofc
4300 Goodfellow Blvd
St. Louis, Missouri 63120

1 Commanding General
U.S. Army Munitions Command
ATTN: AMSMU-RE-R
Dover, New Jersey 07801

1 Commanding General
U.S. Army Tank-Automotive Command
ATTN: AMSTA-CV
Warren, Michigan 48090

1 Commanding General
U.S. Army Weapons Command
ATTN: Sys Anal Ofc
Rock Island, Illinois 61202

Aberdeen Proving Ground

CG, USATECOM
ATTN: AMSTE-TS

AO

CONVERSION OF MICOM, TIME-PHASED, LIFE-CYCLE,
COST-ESTIMATING MOOEL FROM COBOL TO FORTRAN IV

U.S. Army Materiel Systems Analysis Agency (AMSAA), Aberdeen Proving Ground, Maryland 2100S, AMSAA Technical Memorandum No. 131, Report UNCLASSIFIED

CONVERSION OF MICOM, TIME-PHASEO, LIFE-CYCLE, COST-ESTIMATING MODEL FROM COBOL TO FORTRAN IV, Wayne S. Copes, AMSAA; Project No. RDT&E 1P76S801MM11, March 1972

Trace Index Terms

Life-cycle costs
Learning curves
Cost analysis
Economic analysis

1. Life-cycle costs. 2. Learning curves. 3. Cost analysis. 4. Economic analysis.

AD

Copes, Wayne S.

U.S. Army Materiel Systems Analysis Agency (AMSAA), Aberdeen Proving Ground, Maryland 21005, AMSAA Technical Memorandum No. 131, Report UNCLASSIFIED

CONVERSION OF MICOM, TIME-PHASED, LIFE-CYCLE, COST-ESTIMATING MOOEL FROM COBOL TO FORTRAN IV, Wayne S. Copes, AMSAA; Project No. RDT&E 1P76S801MM11, March 1972

Trace Index Terms

Life-cycle costs
Learning curves
Cost analysis
Economic analysis

1. Life-cycle costs. 2. Learning curves. 3. Cost analysis. 4. Economic analysis.

AO

Life-cycle costs

U.S. Army Materiel Systems Analysis Agency (AMSAA), Aberdeen Proving Ground, Maryland 2100S, AMSAA Technical Memorandum No. 131, Report UNCLASSIFIED

CONVERSION OF MICOM, TIME-PHASED, LIFE-CYCLE, COST-ESTIMATING MODEL EROM COBOL TO FORTRAN IV, Wayne S. Copes, AMSAA; Project No. RDT&E 1P76S801MM11, March 1972

Trace Index Terms

Life-cycle costs
Learning curves
Cost analysis
Economic analysis

1. Life-cycle costs. 2. Learning curves. 3. Cost analysis. 4. Economic analysis.

AD

Learning curves

U.S. Army Materiel Systems Analysis Agency (AMSAA), Aberdeen Proving Ground, Maryland 21005, AMSAA Technical Memorandum No. 131, Report UNCLASSIFIED

CONVERSION OF MICOM, TIME-PHASED, LIFE-CYCLE, COST-ESTIMATING MODEL FROM COBOL TO FORTRAN IV, Wayne S. Copes, AMSAA; Project No. RDT&E 1P76S801MM11, March 1972

Trace Index Terms

Life-cycle costs
Learning curves
Cost analysis
Economic analysis

1. Life-cycle costs. 2. Learning curves. 3. Cost analysis. 4. Economic analysis.

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)*

U.S. Army Materiel Systems Analysis Agency
Aberdeen Proving Ground, Maryland

2a. REPORT SECURITY CLASSIFICATION

Unclassified

2b. GROUP

3. REPORT TITLE

CONVERSION OF MICOM, TIME-PHASED, LIFE-CYCLE, COST-ESTIMATING MODEL FROM COBOL TO FORTRAN IV

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

5. AUTHOR(S) *(First name, middle initial, last name)*

Wayne S. Copes

6. REPORT DATE March 1972	7a. TOTAL NO. OF PAGES 88	7b. NO. OF REFS 3

8a. CONTRACT OR GRANT NO.

b. PROJECT NO. RDT&E 1P765801MM11

c.

d.

9a. ORIGINATOR'S REPORT NUMBER(S)

Technical Memorandum No. 131

9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*

10. DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES

12. SPONSORING MILITARY ACTIVITY

U.S. Army Materiel Command
Washington, D.C.

13. ABSTRACT

LICEM is a computer model which may be used to generate Time-Phased Life-Cycle Cost Estimates (LCCE) for personnel or materiel systems. The input to this model is in a form compatible with the Army Materiel Command's Improved Cost Estimating Project, Phase III (ICE-III).

The cost for a system can be computed for as many as thirty equal increments of time, and can be summarized in up to nine levels of complexity. The model estimates a system cost for each time increment as well as the total cost over the life of the system.

The model is written in FORTRAN IV specifically for the Ballistic Research Laboratories' Electronic Scientific Computer (BRLESC).

Unclassified

Security Classification

Unclassified

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Life-cycle costs						
Learning curves						
Cost analysis						
Economic analysis						

Unclassified

Security Classification